Journal of Air Transport Management 112 (2023) 102462

Contents lists available at ScienceDirect

Journal of Air Transport Management

journal homepage: www.elsevier.com/locate/jairtraman

Reproducible air passenger demand estimation

Andreas M. Tillmann [a,c,*], Imke Joormann [b,c], Sabrina C.L. Ammann [a]

[a] Institute for Mathematical Optimization, TU Braunschweig, Germany
[b] Institute of Automotive Management and Industrial Production, TU Braunschweig, Germany
[c] Cluster of Excellence SE²A – Sustainable and Energy-Efficient Aviation, TU Braunschweig, Germany

ARTICLE INFO

Keywords:
Aviation
Air passenger volume
Demand estimation
Gravity models
Machine learning
Data set

ABSTRACT

The availability of passenger demand estimates for air traffic routes is crucial to a plethora of application and research problems ranging from, e.g., optimization of airline fleet utilization to complex simulations of whole air transport systems. However, somewhat surprisingly, such demand estimates appear hard to come by directly or even to generate by means of published models. This is in large parts due to the widespread use of expensive proprietary data (such as airline-specific ticket prices for certain flight connections) which is typically employed both to calibrate demand estimation models as well as to evaluate such models in order to obtain demand estimates for given origin–destination airport pairs. With this work, we propose building a data set for the European air transport system from given base data and automatically extracted and processed data from external sources, all of which are (made) freely available in the public domain and thus enable reproducibility and facilitate comparability of research involving air passenger transportation. Moreover, we challenge the long-standing tradition of calibrating so-called gravity models for demand estimation by standard linear regression. For the European air transport system, using the aforementioned publicly available data, we demonstrate that machine learning models and techniques like neural networks or the "kernel trick" can significantly improve the estimation quality with respect to ordinary least-squares. In fact, our results—the best of which were obtained using our feed-forward neural network model (four hidden layers with tanh and ReLU activations)—achieve a performance at least comparable to what has been reported in earlier works that utilized non-public data. Computer code to generate air passenger demand estimates is made publicly available online along with our base data and implementation to collect and curate external data.

1. Introduction

The estimation of air passenger volumes is an essential aspect of various planning tasks related to air transport systems. For instance, airlines require estimates of future demand in order to decide which flight routes to offer and to determine service frequencies for the routes (Grosche et al., 2007; Vinod, 2021), as well as to set their ticket pricing strategies (Tian et al., 2021). Airport operators include forecasts of passenger throughput in capacity utilization planning and expansion considerations (Suryani et al., 2010). Furthermore, demand forecasts inform the production plans of aircraft manufacturers and are needed by governments for policy decisions (Airports Commission, 2013). Hence, such demand predictions have a widespread impact.

Air passenger demand can be seen from different angles such as passenger numbers or available seat kilometers on certain routes, passenger throughput at airports or air travel growth rates. Most estimation methods build on either time series or explanatory models. In this paper, we are concerned with origin–destination air travel demand in terms of passenger numbers. In particular, we would like the estimation to be able to predict demand for city pairs that are not (yet) connected by air services. Hence, we turn to causal modeling rather than time series analysis; note that the latter relies on historical time series data, which obviously does not exist for unserviced potential routes.

In this context, the most common approach to obtain air passenger volume or (latent) demand estimates between airport or city pairs is the so-called *gravity model*, see, e.g., Verleger (1972), Schultz (1972), Rengaraju and Thamizh Arasan (1992), Grosche et al. (2007), Mao et al. (2015), Terekhov et al. (2015), Zhang et al. (2018), Cook et al. (2017), Hazledine (2017), Becker et al. (2018), Wang and Gao (2021) and references therein. As causal models, gravity models express the passenger volume as a function of socio-economic, demographic and geographic attraction parameters associated with the origin and destination cities as well as the existing air transport system. Specifically, demand is

* Corresponding author.
E-mail addresses: a.tillmann@tu-braunschweig.de (A.M. Tillmann), i.joormann@tu-braunschweig.de (I. Joormann), s.ammann@tu-braunschweig.de (S.C.L. Ammann).

https://doi.org/10.1016/j.jairtraman.2023.102462
Received 26 August 2022; Received in revised form 18 June 2023; Accepted 19 July 2023
Available online 31 July 2023

assumed to be generated as the product of powers of the attraction parameters, see Section 2.1 for a formal description and details. In order to enable a gravity model to estimate demand for a specific area, the model needs to be calibrated, or trained, using available parameter and demand data. Until now, ordinary least-squares regression applied to log-linearized gravity model equations is almost always used for this calibration; see also Section 1.1.

The quality of the resulting predictions depends on several factors that can briefly be summarized as data, features, and market homogeneity. Clearly, availability and reliability of the data used to train the prediction model is a crucial aspect. While, say, population or economic indicator data is often quite readily available from government databases, this is not the case for other data such as prices, route itineraries and, most importantly, actual origin–destination air passenger volumes. Hence, it appears to be common practice in demand estimation research to rely on proprietary booking (and other) data. This is unfortunate because such data can be prohibitively expensive for institutions to acquire, effectively hampering usability of calibrated models as well as reproducibility and comparability of results. However, with the exception of a subset of airline ticketing data from the US market, there apparently do not exist any publicly available sources for air passenger data between true origin–destination city pairs, cf. Grimme and Maertens (2019) and Tian et al. (2021).

Regarding the features, i.e., parameters used as causal variables in the gravity model to explain demands, most studies use statistical arguments to justify which features are included in the final model. In fact, before any such selection of features can even be done, an initial deliberation and definition of parameters that conceivably influence demand needs to take place, which requires a good understanding of the specific problem domain in order to not overlook potentially important features. Furthermore, feature engineering, or design, is used to specify how exactly parameters are to enter the model, i.e., how city-based data is formally combined to route-based attraction parameter values.

Finally, the success of a classical gravity model in terms of overall accuracy may be linked to the homogeneity of the market under consideration, i.e., how comparable the encompassed routes and cities are with respect to social, economic and other indicators (Grosche, 2009; Kanafani, 1983). For instance, a straightforwardly combined model for, say, Europe and South America will generally yield less realistic estimates than separate prediction models for the two continents due to large level disparities in attraction parameters such as distances or buying power.

In the remainder of the present introductory section, we first give an overview of related work on gravity models and other types of air travel demand estimation and forecasting, particularly with regards to the employed mathematical methodologies; see Section 1.1. Furthermore, we touch upon research reproducibility in the context of air transport systems. Then, in Section 1.2, we summarize the contributions of this work, clarify known and potential limitations of our approach and gravity models, and provide an outline of the rest of the paper.

1.1. Related work

We first review relevant prior work on gravity models and their specifics in the air passenger demand estimation context, including parameter specification (explanatory variables) and calibrated model quality assessment, and then discuss time-series based forecasting methods and, finally, aspects of data sources, availability and reproducibility.

Gravity models and their calibration. While gravity models have been extended and modified to account for a variety of circumstances—in particular, adapted to domestic air travel in, e.g., India (Rengaraju and Thamizh Arasan, 1992), Turkey (Sivrikaya and Tunç, 2013) or the United States (Brown and Watkins, 1968; Verleger, 1972), or international flights across the African continent (Adler et al., 2018) or within parts of Europe (Jorge-Calderón, 1997; Grosche et al., 2007; Grosche, 2009)—the basic approach of calibrating a gravity demand estimation model by means of ordinary least-squares regression (OLS, for short) is almost always adhered to. To the best of our knowledge, there are only a few variations of this approach for estimating the type of demand considered in this paper, i.e., air passenger volumes between origin–destination city pairs. Stepwise (OLS-based) regression has been used in order to select which attraction parameters are to partake in the final estimation function, see, e.g., Rengaraju and Thamizh Arasan (1992), although it is known that stepwise regression is unreliable and should be avoided (Smith, 2018; International Civic Aviation Organization, 2006). In Rengaraju and Thamizh Arasan (1992), the authors also employed correlation-based feature design to select whether two parameters for a city pair are combined to a route parameter as sum, product, ratio or sum of reciprocal values. To avoid problems due to zero-values for parameters in their model, Russon (1990), Russon and Riley (1993) modified the underlying multiplicative functional form of the gravity model and used an iterative procedure instead of OLS. Rather than minimizing a least-squares (ℓ_2-norm) term, O'Kelly et al. (1995) suggested minimizing absolute deviations by linear and goal programming. The authors of Terekhov et al. (2015) claimed that gravity models do not take into account possible new flight-connected city pairs and proposed combining OLS with a preceding forecast of the air route network topology; see also Wong et al. (2023).

Gravity models have also been employed for estimating air cargo flows, cf. Aydın and Ülengin (2022) and Baier et al. (2022), and in a vast number of works that investigate various facets of international trade and econometric relations, see, e.g., Kabir et al. (2017), Kepaptsoglou et al. (2010), Anderson (2011) and references therein. Apparently, many econometrics textbooks also advocate the use of OLS to determine gravity model coefficients (cf. Santos Silva and Tenreyro, 2022). However, the seminal paper Santos Silva and Tenreyro (2006) demonstrated that from a statistical estimation viewpoint, OLS is actually a flawed approach in the log-linearized context—it yields an inconsistent estimator and neglects potentially severe issues of heteroskedasticity—and proposed to instead employ a Poisson pseudo-maximum-likelihood (PPML) estimation technique. With several thousand citations to its name, Santos Silva and Tenreyro (2006) clearly had a large impact on trade-flow gravity model research in economics (see also Santos Silva and Tenreyro, 2022), but has gone relatively unnoticed in the air passenger demand estimation context; we are aware of only Zhang et al. (2018), Thou et al. (2018), Oesingmann (2022) and Gelhausen and Berster (2017) employing a PPML approach rather than OLS.

Gravity model parameters. The gravity models found in the literature usually differ with respect to the number and type of attraction parameters. Basic parameters that were employed in virtually all models are population sizes of the cities, flight distances between city-pairs, and gross domestic product numbers at regional or national levels. Other geographical or socio-economic parameters that have been frequently used are airport catchment sizes (e.g., in Grosche et al., 2007) and disposable household incomes (e.g., in Brown and Watkins, 1968); see also Doganis (2019). Many works also included air service-related parameters, say, e.g., ticket prices or flight frequencies (Jorge-Calderón, 1997). From an airline perspective, demand forecasting is intricately linked to the development of pricing strategies (cf., e.g., Tian et al., 2021), and since demand is partially driven by supply, demand estimation models that included ticketing data usually achieved higher accuracy than those without. However, other models intentionally omitted such

parameters, because of the interdependency of air fares and demand and since ticket prices as features can aggravate the model bias toward the existing flight route network structure, which may be inappropriate when the goal is to assess air transport demand for new city pairs for which no pricing data exists, cf. Grosche (2009) and International Civic Aviation Organization (2006).

Furthermore, models designed for specific regions such as those mentioned at the beginning of this subsection often also included attraction parameters that may have a higher local relevance; for instance, Rengaraju and Thamizh Arasan (1992) introduced percentages of literates and university degree holders in their model for India, and Adler et al. (2018) employed indices for corruption levels and risk of violent conflicts in their air passenger demand estimation for Africa. Finally, many gravity models not only included numerical parameters, but also categorical variables that may specify, e.g., whether or not cities share a common first language or lie in the same country (see, e.g., Adler et al., 2018; Mao et al., 2015, respectively).

Gravity model performance assessment. Usually, the quality of the regression or prediction model is judged by the so-called coefficient of determination, R^2, with values ranging widely from quite low (e.g., 0.283 in Grosche, 2009) to high (e.g., 0.952 in Rengaraju and Thamizh Arasan, 1992); recall that R^2 always lies between 0 and 1, with higher values generally indicating a better model fit. However, those values do not allow comparing the associated *models* due to differences in scope (e.g., regional vs. global), selected attraction parameters, and data used for model training. Moreover, note that relatively low R^2 values do not necessarily imply that a model is inappropriate or should be avoided, especially for rather heterogeneous markets and/or when air service parameters are not used, cf., e.g., Grosche (2009). As a real-life example, a few years ago, Canada apparently employed a modified gravity model for air demand estimation whose regression yielded a "reasonably high" R^2 of 0.47 (International Civic Aviation Organization, 2006). Generally, it is unclear how to determine application-specific thresholds for R^2 that would separate good and bad models (James et al., 2021). Despite these and other issues in interpreting R^2 scores—see also Section 4.1, and, e.g., Berk (2004), Anderson and Shanteau (1977) and Birnbaum (1973)—other measures of performance or estimation quality have, to the best of our knowledge, rarely if at all been used in the gravity-model air passenger demand literature thus far, and relatively few studies conducted cross-validation experiments to at least obtain a qualified impression of performance on unseen data.

Time-series based forecasting. Another very common approach to air traffic/demand forecasting involves time-series analysis. Unlike in the causal gravity models, which explain the quantity of interest by expressing it as a function of a variety of influencing variables, time-series methods typically use only the history of the quantity to be predicted to extrapolate trends into the future. Note that, in contrast, causal models for demand estimation are not direct *forecasting* models, but that demand forecasts can be obtained by evaluating a calibrated model on predicted future values of the explanatory variables; see also, e.g., Shmueli (2010) with regards to distinguishing between (temporal) forecasting and explanatory estimation. On the other hand, time-series methods cannot yield predictions for non-existing connections due to the lack of corresponding historical time-series data. Moreover, given the inaccessibility (or expensiveness) of origin–destination demand data, it may be unsurprising that the majority of time-series methods pursued other types of "demand" than the air passenger volumes between city pairs that we consider here, and that many gravity models are designed for. Typical examples of such other demand types are the airport passenger throughput (Sun et al., 2019), air traffic growth rates (Fildes et al., 2011), or aggregate air passenger kilometers (Alekseev and Seixas, 2009; Carmona-Benítez and Nieto, 2020) do model forecasting of passenger flows between city pairs, yet without a causal component and therefore inapplicable to non-existing connections. It would go beyond the scope of this paper to review the many time-series based works on air travel demand prediction, so we would like to refer to the recent surveys (Wang and Gao, 2021; Tian et al., 2021) as suitable starting points into this line of research, where the latter reference emphasizes the connection of demand estimation with airline price and revenue management. Nevertheless, it is worth mentioning that forecasting for other types of demand than origin–destination air passenger volumes has seen a much broader variety of methodologies including, besides the aforementioned time-series methods, system dynamics modeling (Suryani et al., 2010) and, in particular, recent machine learning techniques such as neural network deep learning, see, e.g., Yu (2021), Xiao et al. (2014), Jin et al. (2020), Sun et al. (2019) and Alekseev and Seixas (2009), or hybrid approaches utilizing time-series data of several explanatory variables for predictions with neural networks, cf. BaFail (2004). As we will demonstrate later, machine learning can also successfully improve origin–destination demand prediction.

Data availability issues. Finally, it is important to note that data sources across different papers are diverse and nearly always include proprietary booking data sets, called market information data tapes or similar, offered by Sabre, Amadeus, OAG, and others. This prevents any comparison or reassessment of existing models unless one is able and willing to pay the steep prices—up to thousands of US dollars—to obtain such data sets. We are only aware of very few efforts to address the current lack of reliable, publicly available data and reproducibility of research results in the context of air transport systems. An early contribution in this area was the generic cost function for aircraft trips derived in Swan and Adler (2006), which allows to estimate operating costs based on aircraft capacity and flight length. The composition of the cost function is described in detail, as is the data used for its calibration. However, this data does not appear to be directly or freely available, although evaluation of the final function does not require knowledge of it. Similarly, an estimate of air fares (ticket prices) as a function of oil prices and flight distance has been provided in Terekhov (2017). This may be very useful since historical oil prices are much easier to come by from public databases than average ticket prices, which would typically need to be gathered over long periods of time by crawling, e.g., online travel agents as done in Bilotkach and Pejcinovska (2012). The very recent work Förster et al. (2022) gave further estimation functions for a variety of aircraft operating costs, including, in particular, CO_2-emissions and fuel burn. Somewhat more loosely related, Park and O'Kelly (2016) proposed a mathematical model to disaggregate air passenger flows on route segments, i.e., to disentangle traveler volume on flights in order to estimate the true passenger origin–destination matrix; recall that passenger itineraries and, hence, the *actual* origin–destination data are generally not publicly available. Furthermore, it is also worth mentioning that for a subset of European countries, Angelova and Blanco Lupio (2020) have used publicly available data to compile a database of climate indicators such as temperatures and precipitation levels aligned to the highest-resolution statistical regions used by the European Union, making it compatible with the wealth of data freely available from Eurostat databases. Such indicators may be and have sometimes been used in gravity models as explanatory variables for demand prediction as well. Finally, we remark that the papers Huang et al. (2013), Mao et al. (2015) provide some interesting discussion on data availability as well as gravity model setups; they claim to provide an open-access data repository on annual and monthly global flows of air travel passengers, respectively, but the given URL links to their data are not functional, and we could not locate these data sets anywhere else.

1.2. Contributions and limitations

Our goal is to estimate the air passenger volume/demand between arbitrary European city pairs, using only reliable and update-capable

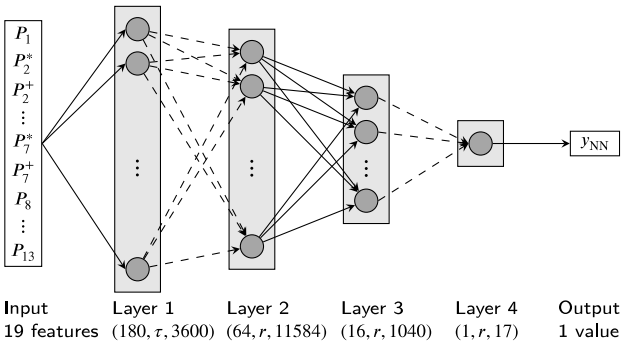

Fig. 1. Schematic illustration of the employed neural network architecture; for each hidden layer, we state its width (number of neurons), activation function, and trainable weight parameters. Dashed connections are subject to dropout regularization.

publicly available data sources to ensure reproducibility and to enable future research that depends on this data, e.g., air transport system planning problems. To this end, we propose a methodology to build a data set by extracting and curating data from freely accessible and actively maintained online sources, and compare different approaches to handle the calibration of a gravity model (and two other models); we also touch upon feature selection. In addition to the prevalent OLS method and variants of linear regression, we also evaluate the PPML approach and some popular machine learning techniques, namely kernelization, support vector regression and deep learning with neural networks. Our computational results show that competitive prediction quality can be achieved based purely on freely available data, i.e., without relying on expensive proprietary data sets. In particular, we demonstrate that modern machine learning methods are superior to OLS as well as PPML, and hence may generally improve estimation accuracy in the present context, irrespective of the available data set. Overall, our neural network model (see Section 2.2.2 and Fig. 1 for the architectural specifics) showed the best average performance and, with very high statistical confidence, significantly outperforms all other approaches. Furthermore, we discuss feature selection and design, and show that mixed-integer programming offers a competitive, efficient and more flexible alternative to standard approaches. To advance reproducibility and comparability, we make our base data and easily extensible implementation (code) of our external-data collection methodology and different models/calibration algorithms available online in an open-source repository.

To predict the air passenger demand for city pairs where air services do not (yet) exist, gravity models have been deemed a suitable approach, despite their arguably somewhat simplistic form that might preclude applicability to more than coarse estimation (cf., e.g., International Civic Aviation Organization, 2006; Nõmmik and Kukemeld, 2016; Cleophas et al., 2009; Wong et al., 2023). This suitability also holds for more general causal models such as, e.g., the neural network we employ, although they may be more difficult to interpret than a simple gravity model due to their generally more opaque or black-box nature. We do not include service-related attraction parameters in our models, in order to avoid increasing the bias toward existing flight routes and market biases, cf. Grosche (2009), and due to the difficulty in reliably and reproducibly obtaining values such as ticket prices from public databases. Nevertheless, we train and validate our prediction models only for cities with one or more airports, because otherwise (ground-truth passenger volume) data is non-existent or one could not evaluate the model accuracy, respectively. As noted in earlier studies (see, e.g., Mao et al., 2015), this unavoidably introduces some bias and inaccuracy related to the existing route network, since the available training data only pertains to passengers on direct flights. As the model cannot distinguish true origin–destination pairs, it does not account for transfer passengers from multi-leg flights, which also implies that passenger numbers on the considered connections are not necessarily truly independent. Some of these issues may be resolved when commercial data sets (e.g., trip itineraries from market information data tapes) are employed, but this would defeat the purpose of using only publicly and freely available data. Nonetheless, the negative distorting impact on the estimation results in our specific case should be very small, since intra-European trips are predominantly non-stop direct flights (cf. Suau-Sanchez et al., 2017; Dobruszkes, 2013).

Considering that actual origin–destination demand data for Europe is unavailable (at least from public sources), we use the numbers of passengers on-board flights between European cities as a proxy for the unknown true demand. This data pertains to bidirectional travel—i.e., we do consider city-pair connections but do not distinguish which direction passengers travel in—and is available from the Eurostat public database. In fact, this database is the main external data source for the parameter data used in this paper; more details on data sourcing will be given in Section 3. Our data contains parameter aggregates for a whole year, and consequently, the estimation models predict yearly air passenger volumes. Predictions for different periods (holidays, summer vacation, etc.) require more fine-grained input, which is simply a matter of definition—the models themselves remain the same, but would need to be recalibrated, and Eurostat data is often available on a monthly basis as well, including passengers-on-board data. The specific "log–log" form of the gravity model we employ (see Section 2.1) is generally considered the most appropriate for aggregate traffic demand estimation (International Civic Aviation Organization, 2006). Nevertheless, with the criticism of this log-transformation from Santos Silva and Tenreyro (2006) in mind, we note that we also perform calibration directly on the multiplicative gravity formula (using PPML) and without any underlying model-driven functional relationship (i.e., training a neural network). The attraction parameters we employ are mostly fairly standard, but may still involve some simplifications. For instance, the catchment area of an airport is typically based on travel times to an airport or similar local factors; for convenience, we define a catchment area based on the Eurostat statistical regions the airport falls into, cf. Section 3.

It is important to emphasize that one should always make use of the best data one has access to. Here, we focus on treating the European air system as a whole, but note that since applications may call for different accuracy requirements, model extensions or market segmentation, our methods should be further specialized before using the resulting demand predictions in sensitive applications. In particular, airlines, for which high-quality demand prediction is of core value to their business and plays an important role in revenue management systems, would certainly utilize at least their own booking data and typically further independent variables relating to, e.g., ticket prices. In this paper, we do not distinguish between airlines but consider the (European) air route network as a whole, and use only publicly available data to enable at least coarse estimations when access to proprietary data is unavailable. (Nevertheless, since our models and implementation could be easily extended to incorporate further parameters and/or data sources, they might in principle also be adopted by airline managers with access to better data.) Moreover, it should be mentioned that the structure of the neural network we employed has not been optimized specifically for the task at hand. Therefore, while our computational results provide a proof of concept that a neural network model can significantly outperform all prevalent other approach in this work's context, a rigorous neural architecture search would be a logical step before actually applying demand estimations from such a model. A similar remark applies to the other machine learning techniques, where variations with respect to, in particular, the selected kernel functions are possible and may lead to further improvements.

The remainder of this paper is organized as follows. In the next section, we first introduce the main gravity model, define its attraction parameters and describe its calibration by means of OLS and variants,

Table 1
Overview of features/attraction parameters used in the models.

Feature	Description	Type
P_1	Great-circle distance in kilometers	Numerical
P_2^*, P_2^+	Population (product, sum)	
P_3^*, P_3^+	Catchment area population (product, sum)	
P_4^*, P_4^+	Regional gross domestic product (product, sum)	
P_5^*, P_5^+	National price level indices (product, sum)	
P_6^*, P_6^+	Nights spent in tourist accommodations (product, sum)	
P_7^*, P_7^+	Poverty risk percentage (product, sum)	
P_8	Same currency	Categorical/indicator
P_9, P_{10}	Domestic and international connections (complementary)	
P_{11}	Coastal location involvement	
P_{12}	Intra-EU connection	
P_{13}	Island location involvement	

including feature selection and kernelization (Section 2.1). Then, we introduce the more advanced machine learning approaches of kernelized support vector regression and neural networks (Section 2.2), and finally the PPML estimator (Section 2.3). In Section 3, we give details on our proposed data set generation, namely the base data we make publicly available along with our code to collect and process data from external sources, and which sources were used. Section 4 first provides some details on our implementation of the various estimation methods and the measures used to assess and compare the trained models, before presenting and discussing our computational results; some details are deferred to Appendices A and B. The final Section 5 summarizes our findings and gives some pointers to possible extensions, refinements and future research.

2. Modeling air passenger demand

We follow the widespread approach of so-called *gravity models* to generate passenger demand estimates. Unfortunately, as alluded to earlier, virtually all such models from the literature involve parameters that are not readily and publicly accessible, e.g., airline ticketing data and other air service characteristics. Moreover, while simpler gravity models involving only general economic and geographical characteristics have been proposed as well (see, e.g., Grosche et al., 2007; Grosche, 2009), the calibration processes of these models were still performed using proprietary booking data and are thus not directly reproducible.

We take inspiration from previous approaches (cf. the discussion in Section 1) and set up our own gravity model for origin–destination air passenger volume estimation; additionally, we will build a neural network model using the same input parameters. Rather than relying on opaque and publicly unavailable data, our models will be calibrated using only freely accessible data from reliable sources, see Section 3.

2.1. Gravity model

Gravity models are causal models that generally aim to explain the quantity of interest as a function of several influencing parameters or variables. The general form of a gravity model to estimate (bidirectional) passenger demand $d_{ij} = d_{ji}$ between a city pair (i, j) is as follows:

$$d_{ij} = e^{\beta_0} \prod_{k=1}^{K} P_k(i,j)^{\beta_k}, \tag{1}$$

where $e \approx 2.71828$ is Euler's constant, $P_1(i,j), \dots, P_K(i,j)$ are the so-called *attraction parameters* that influence the prediction, and the exponents $\beta_0, \beta_1, \dots, \beta_K$ are obtained by the model calibration (training or regression). The standard calibration technique is to take the logarithm of the gravity equations (1) evaluated at given data points and perform an ordinary least-squares regression to obtain the exponent values, see Section 2.1.2; variants of this approach will be described in Sections 2.1.3 and 2.1.4.

2.1.1. Model input (attraction) parameters

The parameters $P_k(i,j)$ can either be properties of the route between cities i and j (e.g., the flight distance) or they can be derived from parameters that are associated with each city separately. Both numerical values, such as populations, as well as categorical variables are admissible. The latter, sometimes also called "dummy" variables, are indicators for certain properties such as whether i and j share the same language. As mentioned earlier, there are several common ways to combine individual-city parameters to city-pair attraction parameters (cf., e.g., Rengaraju and Thamizh Arasan, 1992); we focus on the sum and the product of two numerical values associated with i and j individually, and denote the corresponding attraction parameters and exponents as $P_k^+(i,j)$, β_k^+ and $P_k^*(i,j)$, β_k^*, respectively. In fact, we will also look at using *both* the product and sum parameters; while this may naturally improve the final model fit (as including additional explanatory variables always does), it may also introduce some correlation issues into the model that may need to be weighed against any potential gains. Later, in the context of feature selection, we will, in particular, investigate allowing only one of the respective two definitions to be used in the model, and thereby automating the combined choices of definitions for all affected attraction parameters that lead to the best results. Categorical parameters are usually obtained by logical operations on the individual-city indicators, e.g., a city-pair parameter may be set to "true" if and only if both individual-city indicators are.

In order to precisely define attraction parameters, one first needs to decide on the underlying basic data resolution, i.e., which geographical areas are used to obtain economic and socio-demographical data such as regional price level indices or population numbers. To some extent, this decision depends on the availability of the data intended for model calibration and estimation. Thus, for the purposes of this paper, we will utilize the statistical units defined by the European Union called, from smallest to largest, *NUTS-3*, *NUTS-2* and *NUTS-1* regions, cf. Eurostat (European Commission) (2021). For example, the population of a city will then be measured as the population of the NUTS-3 regions the city or its metropolitan area—also as defined by Eurostat—is part of; for large cities, this can encompass multiple NUTS-3 regions.

For easier reference, we summarize the employed features in Table 1. We base this initial selection of parameters on the existing literature, i.e., most if not all attraction parameters we employ have been used in previous studies such as those mentioned in Section 1. Since the precise definitions are typically different, we provide the details of ours in the following. Here, we consider the following numerical attraction parameters:

- *Distance*: $P_1(i,j)$ is the great-circle distance in kilometers between the airports of cities i and j. In case of multi-airport metropolitan or NUTS-3 regions, distances are averaged over the associated airports.
- *Population*: $P_2^*(i,j) := \mathcal{P}_i\mathcal{P}_j$ and $P_2^+(i,j) := \mathcal{P}_i + \mathcal{P}_j$, where $\mathcal{P}_i$ is the population of the statistical region in which city i is located. Statistical region here means either, if applicable, the metropolitan area of city i, or else, the NUTS-3 region containing it.

- *Catchment*: $P_3^*(i,j) := C_i C_j$ and $P_3^+(i,j) := C_i + C_j$, where C_i is the population of the "catchment area" of city i (or its airport(s)). Catchment areas are usually defined via fixed distances or driving times to the respective airport, see, e.g., Mao et al. (2015) and Grosche et al. (2007). However, reliably and consistently collecting data based on such definitions can be a cumbersome task, since driving times can vary significantly depending on local infrastructure, and it is unclear how population numbers within a certain distance radius from an airport should be obtained. Therefore, we choose as catchment area for a city i the next-larger statistical unit(s) encompassing the NUTS-3 region(s) containing i, i.e., the (combined) NUTS-2 regions.
- *Regional Gross Domestic Product (GDP)*: $P_4^*(i,j) := G_i G_j$ and $P_4^+(i,j) := G_i + G_j$, where G_i is the gross domestic product (in Million Euros) of the NUTS-3 region(s) or metropolitan area associated with i.
- *National Price Level Index (PLI)*: $P_5^*(i,j) := L_i L_j$ and $P_5^+(i,j) := L_i + L_j$, where L_i is the price level index of the country containing city i with respect to the baseline 100, which corresponds to the average PLI of the 27 EU nations (in the chosen year).
- *Nights stayed at tourist accommodations*: $P_6^*(i,j) := N_i N_j$ and $P_6^+(i,j) := N_i + N_j$, where N_i is the annual number of nights guests spent in hotels, holiday apartments, campgrounds and guest houses located in the NUTS-2 region(s) covering the NUTS-3 regions or metro area associated with city i; this serves as a measure for tourism or business center popularity.
- *Poverty risk*: $P_7^*(i,j) := R_i R_j$ and $P_7^+(i,j) := R_i + R_j$, with R_i being the percentage of the population of the country containing i that is at risk of poverty or social exclusion.

Also, we utilize the following categorical parameters:

- *Same currency*: $P_8(i,j)$ is "true" if and only if the countries of i and j share the same currency. Note that for our selection of countries (see Section 3.1), this coincides with whether both countries use the Euro.
- *Domestic*: $P_9(i,j)$ is "true" whenever i and j lie in the same country, and "false" otherwise.
- *International*: $P_{10}(i,j)$ is "true" whenever i and j lie in different countries, and "false" otherwise. Note that P_9 and P_{10} are complementary, so that the standard model bias term e^{β_0} is, in fact, not needed in our concrete model and will indeed be omitted.
- *Coastal location*: $P_{11}(i,j)$ is "true" whenever at least one of i and j lies in a coastal region (i.e., a NUTS-3 region that has a sea coast), and "false" otherwise.
- *Intra-EU*: $P_{12}(i,j)$ is "true" if and only if both i and j lie in one of the 27 EU member countries. Note that 19 EU countries use the Euro, so that $P_8(i,j)$ being "true" implies $P_{12}(i,j)$ is necessarily also "true" but not vice versa.
- *Islands*: $P_{13}(i,j)$ is "true" if and only if at least one of i and j lies on an island (as defined by Eurostat).

The "true/false" or "on/off" nature of categorical features is realized by assigning the two distinct values e (true) and 1 (false). Since the respective logarithms are 1 and 0, this correspondingly includes or excludes these parameters' contributions for certain data points during the model calibration process, see Section 2.1.2 below. For models without log-linearization—in particular, the neural network model from Section 2.2.2—we use binary feature values directly.

More details on how we collect and aggregate the data will be provided later in Section 3. At this point, we briefly mention that the (training) data set we work with contains nearly 4000 data points associated with over 300 (aggregated) airports from 33 European countries.

2.1.2. Standard calibration method: Ordinary least-squares/linear regression

The predominant calibration process for gravity models is the ordinary least-squares (OLS) approach, which amounts to a simple linear regression: Taking the natural logarithm of the gravity formula (1) yields

$$\log(d_{ij}) = \beta_0 + \sum_{k=1}^{K} \log(P_k(i,j))\beta_k, \tag{2}$$

which gives rise to a (typically overdetermined) linear system of the form

$$\hat{P}\beta = \hat{d}, \tag{3}$$

where $\hat{P} \in \mathbb{R}^{N\times(K+1)}$ and $\hat{d} \in \mathbb{R}^N$ are given and the coefficient vector $\beta \in \mathbb{R}^{(K+1)}$ is to be determined; here and henceforth, N denotes the number of data points available for model calibration, and K is the number of input features. Assuming the columns of $\hat{P}$ are linearly independent (otherwise the model contains dependent features) and $N > K$, we can then obtain the OLS solution

$$\hat{\beta} := \arg\min_{\beta} \|\hat{P}\beta - \hat{d}\|_2^2 = (\hat{P}^\top \hat{P})^{-1} \hat{P}^\top \hat{d}.$$

The calibrated gravity model to be used for demand prediction then amounts to (1) with $\beta = \hat{\beta}$.

We remark that with the attraction parameters defined in Section 2.1.1, our gravity model is concretely specified via

$$d_{ij} = P_1(i,j)^{\beta_1} \prod_{k=2}^{7} P_k^+(i,j)^{\beta_k^+} P_k^*(i,j)^{\beta_k^*} \prod_{k=8}^{13} P_k(i,j)^{\beta_k}.$$

Recall that, by definition, $\log(P_k(i,j)) \in \{0,1\}$ for $k = 8,\dots,13$, resulting in the associated respective coefficient β_k being either present (as a summand) in the logarithmic equation or not, for each pair (i,j). Moreover, we observe that $\beta \in \mathbb{R}^{19}$ in our concrete case, if all $K = 19$ features are retained, and we do not have a bias term β_0 due to the complementarity of parameters $P_9(i,j)$ and $P_{10}(i,j)$. (A constant bias term would be linearly dependent on these two features, since $\log(P_9(i,j)) + \log(P_{10}(i,j)) = 1$ for all (i,j), and therefore is redundant.) For notational convenience, we will henceforth not distinguish between the general form (with bias term) and our concrete model, and refer to either, as in (3), with data $\hat{P} \in \mathbb{R}^{N\times K}$ and $\hat{d} \in \mathbb{R}^N$ and variables $\beta \in \mathbb{R}^K$ in our case.

Typically, calibrated gravity models are analyzed by computing one or more regression quality scores—most commonly, the coefficient of determination R^2—as well as the so-called p-values (of t-statistics) associated with the explanatory variables β_k to illuminate whether the individual attraction parameters are statistically significant, see, e.g., Cook et al. (2017), Adler et al. (2018) and Schultz (1972). Further statistical characteristics are occasionally also considered, see, e.g., Jorge-Calderón (1997) and Grosche et al. (2007). However, the significance of explanatory variables is, in fact, a core question of the broader field of *feature selection*—also called variable or best subset selection, cf., e.g., Hastie et al. (2009) and James et al. (2021)—and can be handled in various ways, some of which we briefly touch upon in the following. We will discuss R^2 and further notions of model accuracy later, see Section 4.1, and analyze feature selection in more detail in Section 4.2.2 and Appendix B.

2.1.3. Variants of linear regression and feature selection

Some variants of the classical OLS approach have also been adopted in the context of gravity models for air passenger volume estimation. In particular, faced with uncertainty regarding the relevance of attraction parameters in the prediction model, some works resort to a step-wise regression that iteratively adds or removes parameters based on reoptimization and updating p-values, which indicate statistical significance. For air passenger demand estimation, this was done, e.g., in Rengaraju and Thamizh Arasan (1992). However, generally, step-wise regression

is *not* a reliable method, as laid out in Smith (2018) (see also International Civic Aviation Organization, 2006 and our discussion in Appendix B). To instead combine the OLS approach directly with a mechanism to possibly discard less relevant attraction parameters, we resort to the now well-established feature selection by *sparse regression* model that adds a cardinality constraint to the underlying least-squares problem:

$$\min_{\beta} \|\hat{P}\beta - \hat{d}\|_2^2 \qquad \text{s.t.} \quad \|\beta\|_0 \leq s, \tag{4}$$

where $\|\beta\|_0 := |\{k : \beta_k \neq 0\}|$, $s \in \mathbb{N}$. Here, the sparsity level s needs to be chosen a priori, i.e., one needs to limit the number of features/parameters explicitly. For a given s, the features that yield the best data fit (with respect to the objective of the OLS approach) are then identified automatically by solving the optimization problem (4). In the present context, we will explore all sparsity levels in order to obtain a complete picture of the best feature choices and to identify a "transition" point, i.e., a number of parameters beyond which including more does not significantly improve the model fit any further. The sparse regression problem (4) can be written as the mixed-integer quadratic program

$$\begin{aligned} \min_{\beta,y} \quad & \beta^\top(\hat{P}^\top\hat{P})\beta - 2\hat{d}^\top\hat{P}\beta \\ \text{s.t.} \quad & -My \leq \beta \leq My, \quad \mathbb{1}^\top y \leq s, \\ & y \in \{0,1\}^K, \ \beta \in \mathbb{R}^K, \end{aligned} \tag{5}$$

where $M > 0$ is a so-called *Big-M* constant. The constraints $-My_i \leq \beta_i \leq My_i$ force β_i to zero if $y_i = 0$, but for sufficiently large M, they pose no restriction on β_i if $y_i = 1$, so that the constraint $\mathbb{1}^\top y \leq s$ enforces a cardinality of β of at most s; see, e.g., Tillmann et al. (2021) for details and alternative reformulations. Although generally NP-hard, (5) can be solved efficiently with off-the-shelf MIP solvers such as SCIP (Bestuzheva et al., 2021), CPLEX (IBM ILOG, 2022), or Gurobi (Gurobi Optimization, LLC, 2022) in the dimensions encountered in the present work.

Moreover, we can additionally equip the cardinality-constrained least-squares problem (4) in its form (5) with constraints $y_{2\ell} + y_{2\ell+1} \leq 1$, $\ell \in \{1,2,3\}$, that allow only one of each pair $(P_k^*(i,j), P_k^+(i,j))$, $k \in \{2,\dots,7\}$, to be actively used in the model. Recall that such route-feature pairs are derived from the same city-features (e.g., population) and hence might introduce undesired correlation issues into the model or be regarded as questionable feature design. This can be avoided by the above extra constraints, which let the model automatically identify the best combination of corresponding attraction parameter design choices; see also the computational experiments in Section 4.2.3.

For the sake of completeness, we include two stepwise regression schemes; our experiments later will confirm previous skepticism about this kind of approach (cf. Smith, 2018) and show that, in our application, it does not offer any advantage over sparse regression for feature selection. The methods we employ here are *Recursive Feature Elimination* (RFE) and *Sequential Forward Selection* (SFS), both with cross-validation. The former attempts to select the best model parameters by iteratively evaluating the OLS model on a current subset of features (using cross-validation), starting with all parameters and consecutively removing the "least important" feature, see Guyon et al. (2002); thus, it decides on the number of features as well as their selection. The SFS method iteratively adds the feature most improving a cross-validation score in a greedy fashion, up to a specified maximal number of features, cf. Ferri et al. (1994). Since RFE and SFS base their selection on cross-validation scores, we later introduce a modified version of the sparse regression problem (4) that also takes data splits into account, see Section 4.2.2 for the details.

Furthermore, in addition to the standard least-*squares* (OLS) approach, we also consider the so-called *least absolute value* (LAV) regression

$$\min \|\hat{P}\beta - \hat{d}\|_1, \tag{6}$$

which can be reformulated and solved as a linear program (cf., e.g., Tillmann et al., 2021). The intuition behind this method is trying to find a linear regression function that has a small number of data points that are not fit well, as opposed to trying to evenly spread the approximation error as in the OLS approach; see Dielman (2005) and O'Kelly et al. (1995) for details. Note that LAV regression, as well as the next methods we consider here, could also be combined with feature selection mechanisms (e.g., (6) with cardinality constraints as in (4) could be reformulated and solved as a mixed-integer linear program); however, for the sake of overall clarity, we will not pursue such extensions in the present paper.

Finally, it is worth mentioning that should the data contain severe outliers, one could apply robust regression techniques; see, e.g., Rousseeuw and Leroy (1987) as a starting point. However, since such outliers do not seem to be present in our data set, we do not go into the details here.

2.1.4. Kernel ridge regression

Last but not least, we can adopt a very popular machine learning approach and embellish the OLS model with a *nonlinear kernel*, thereby extending the representable functions of such models quite significantly. While the gravity model in its logarithmic form (2) is linear, the actual causal relationship between the (logarithmized) air passenger demand volumes and explanatory parameters may, in fact, not be linear. Thus, it seems quite natural to explore the so-called *kernel trick* in the present context, and perform a *kernel ridge regression* (cf., e.g., Schölkopf and Smola, 2001; Vovk, 2013). Briefly, a (positive definite) kernel is a function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is the domain of input data vectors, satisfying $\kappa(x,x') \mapsto \langle \Phi(x), \Phi(x') \rangle = \Phi(x)^\top \Phi(x')$ for all $x, x' \in \mathcal{X}$, where $\Phi : \mathcal{X} \to \mathbb{R}^D$ maps the input vectors into the so-called feature space, which is typically very high-dimensional. Thus, a kernel can be interpreted as a similarity measure in feature space. The Gram matrix associated with κ and inputs $x^1, \dots, x^N \in \mathcal{X}$ is given by $\mathcal{K} := (\kappa(x^i, x^j))_{ij}$. Since it is positive definite, its eigendecomposition is $\mathcal{K} = U^\top \Lambda U$ with a diagonal matrix Λ of eigenvalues $\Lambda_{jj} > 0$, and it holds that $\Phi(x^j) = \Lambda^{1/2} U_{\cdot j}$, where $U_{\cdot j}$ is the jth column of U (i.e., the jth eigenvector of $\mathcal{K}$). Hence, with such a feature map $\Phi : \mathcal{X} \to \mathbb{R}^D$, we can state our kernel ridge regression problem as

$$\min_{\beta} \|\Phi(\hat{P})\beta - \hat{d}\|_2^2 + \alpha\|\beta\|_2^2, \tag{7}$$

where we abbreviate $\Phi(\hat{P}) := (\Phi(\hat{P}_{1\cdot}), \dots, \Phi(\hat{P}_{N\cdot}))^\top \in \mathbb{R}^{N \times D}$, with $\hat{P}_{i\cdot} \in \mathbb{R}^K$ denoting the ith row of $\hat{P}$, i.e., the ith input feature vector. Note that here, $\beta \in \mathbb{R}^D$ since the regression now takes place in the feature space, which may be much larger than the original space ($\mathbb{R}^K$) or even infinite-dimensional, cf. Murphy (2012). Hence, including a feature map generalizes OLS; the ridge penalty term $\alpha\|\beta\|_2^2$ prevents overfitting, especially if the feature space is extremely high-dimensional. Analogously to OLS, (7) admits a closed-form expression for its optimal solution (if $D < \infty$), namely

$$\begin{aligned} \hat{\beta} &= \arg\min_{\beta} \|\Phi(\hat{P})\beta - \hat{d}\|_2^2 + \alpha\|\beta\|_2^2 \\ &= \left(\Phi(\hat{P})^\top \Phi(\hat{P}) + \alpha I\right)^{-1} \Phi(\hat{P})^\top \hat{d}, \end{aligned}$$

where I is the $D \times D$ identity matrix. In practice, however, the crucial aspect of kernelization—the actual "kernel trick"—exploits the observation that the dual problem of (7) can be solved in a such way that only access to inner products of vectors in the feature space is ever needed. Since these inner products of the form $\Phi(x)^\top \Phi(x')$ are, by construction, equal to kernel evaluations $\kappa(x,x')$, the corresponding algorithms hence only require the kernel function and never need to perform computations with (potentially infinite-dimensional) vectors $\Phi(x)$ at all, retaining the enhanced expressiveness of working in a higher-dimensional space *implicitly*. Indeed, neither Φ nor the D-dimensional solution vector $\hat{\beta}$ need to be used explicitly even when applying the trained model. It goes beyond the scope of this paper

to fully describe how this is done; we refer the interested reader to, e.g., Murphy (2012), Hoffman et al. (2008), Schölkopf and Smola (2001) and Hastie et al. (2009) for details on the kernel trick and its utilization in kernel ridge regression and several other machine learning models. For the kernel ridge regression in the present work, we use $\alpha := 0.9$ and the radial basis function (RBF) kernel $\kappa(\hat{P}_{i\cdot}, \hat{P}_{j\cdot}) := e^{-\|\hat{P}_{i\cdot} - \hat{P}_{j\cdot}\|_2^2/K}$.

2.2. Advanced machine learning models

Assuming a nonlinear (and not necessarily log-linear) causal relationship between explanatory variables, i.e., attraction parameters, and air passenger volumes, it seems only natural to move from standard linear regression to more powerful prediction methods. The explosion of research work on machine learning in the recent past, with thousands of papers published each year, makes it impossible to provide a comprehensive survey in the scope of this paper. Therefore, we will focus on two well-known and relatively straightforward machine learning tools to serve as examples of these approaches that, surprisingly, have been largely neglected in the context of causal origin–destination air passenger demand estimation thus far: support vector machines and neural networks (deep learning). (Recall that neural networks have found some application in air traffic demand forecasting based on time series, cf. Section 1.1, but that those methods are not applicable here.)

2.2.1. (Kernelized) support vector regression

The first machine learning approach we will employ for air passenger demand prediction is *support vector regression* (SVR), see, e.g., Smola and Schölkopf (2004). Similarly to kernel ridge regression, we directly incorporate a nonlinear kernel $\kappa(\hat{P}_{i\cdot}, \hat{P}_{j\cdot}) = \Phi(\hat{P}_{i\cdot})^\top \Phi(\hat{P}_{j\cdot})$ with associated feature map $\Phi : \mathbb{R}^N \to \mathbb{R}^D$, so that the generalized SVR problem we consider can be written as

$$\begin{aligned} \min_{\beta} \quad & \frac{1}{2}\|\beta\|_2^2 + \frac{1}{\mu}\sum_{i=1}^{N}\left((\xi_i^-)^\ell + (\xi_i^+)^\ell\right) && (8)\\ \text{s.t.} \quad & -\epsilon - \xi_i^- \le \Phi(\hat{P}_{i\cdot})^\top\beta - \hat{d}_i \le \epsilon + \xi_i^+ \qquad \forall i \in \{1,\dots,N\} \\ & \beta \in \mathbb{R}^D;\ \xi^-, \xi^+ \in \mathbb{R}^N_{\ge 0}. \end{aligned}$$

In this model, absolute violations of the (feature-space) linear equations $\Phi(\hat{P})\beta = \hat{d}$ up to a value of $\epsilon \ge 0$ are considered admissible, i.e., any feasible solution $\tilde{\beta}$ with $\|\Phi(\hat{P})\tilde{\beta} - \hat{d}\|_\infty \le \epsilon$ incurs no penalty. By introducing the slack variables $\xi^\pm$, larger violations are still tolerated, but discouraged by the penalty term in the objective function, whose strength is controlled by the parameters $\mu, \ell > 0$. Note that SVR is a generalization of kernel ridge regression: by setting $\ell = 2$, $\alpha = 1/(2\mu)$ and $\epsilon = 0$, and with some obvious rearranging, one can obtain (7) as a special case of (8), cf. Vovk (2013).

For our experiments, we set $\ell := 1$ and $\epsilon := 0.2$, and again use the RBF kernel. Linearly penalizing the slack variables implies that we are optimizing an *ϵ-insensitive loss function* (Smola and Schölkopf, 2004) with respect to data fidelity in the feature space. Such a function is defined as

$$|x|_\epsilon := \begin{cases} 0, & \text{if } |x| \le \epsilon, \\ |x| - \epsilon, & \text{otherwise,} \end{cases}$$

applied componentwise and summing up the results for vector arguments. Substituting $\alpha = 1/(2\mu)$, the kernel SVR model (8) can then be written as

$$\min_{\beta} \left|\Phi(\hat{P})\beta - \hat{d}\right|_\epsilon + \alpha\|\beta\|_2^2.$$

Thus, the difference to kernel ridge regression is that the ℓ_2-norm data fidelity term is replaced by a generalized ℓ_1-norm. Analogously to the kernel ridge regression, in practice, one can completely avoid explicitly handling potentially infinite-dimensional feature space operations by solving the dual problem, which again can be done using only inner products $\Phi(\hat{P}_{i\cdot})^\top\Phi(\hat{P}_{j\cdot})$ that can be replaced by kernel evaluations $\kappa(\hat{P}_{i\cdot}, \hat{P}_{j\cdot})$, see, e.g., Smola and Schölkopf (2004), Murphy (2012) and Chang and Lin (2011) for details.

2.2.2. Neural network

Neural networks have become a staple in various pattern recognition, classification and prediction applications due to their ability to approximate, in principle, arbitrary unknown functions given a suitable network architecture and sufficient data on which to train the network parameters. Detailed introductions to the topic of deep learning with neural networks can be found, e.g., in Murphy (2012), Goodfellow et al. (2016) and James et al. (2021). For simplicity, we employ a fully connected feed-forward neural net—also known as a *multilayer perceptron*—with four hidden layers and trained with dropout regularization (cf. Srivastava et al., 2014). In the first layer, we use hyperbolic tangent activations $\tau(x) := \tanh(x)$, applied componentwise to vector arguments, and for the remaining three layers, we employ rectified linear units (ReLU functions) $r(x) := \max\{0, x\}$, again applied componentwise for vector inputs. This network layout was found to work quite well for our purposes, though better neural architectures likely exist—we manually tried out several layouts instead of performing an automated extensive search for a good layout, which typically is extremely time-consuming, cf. Elsken et al. (2019), and hence was out of scope for this paper. An overview of our network structure is given in Fig. 1; in total, this neural net has 16 241 trainable weights.

The trained neural net provides a prediction $y_{\mathrm{NN}}^{(i,j)}$ for the air passenger volume or demand between a city pair (i, j) based on an input vector $p^{(i,j)} \in \mathbb{R}^{19}$ of route-based features—the same as used in the other models—via

$$y_{\mathrm{NN}}^{(i,j)} := h^4\Big(W^4 h^3\big(W^3 h^2\big(W^2 h^1\big(W^1 p^{(i,j)} + b^1\big) + b^2\big) + b^3\big) + b^4\Big),$$

where h^ℓ and (W^ℓ, b^ℓ) are the activation function and weight parameters of the ℓ-th hidden layer, $\ell = 1, \dots, 4$. Thus, in a fully connected neural net, every neuron in every layer receives as input the output of the previous layer, and the layer output concatenates the outputs of each neuron, i.e., the results of the respective activation function applied componentwise to an affine transform of the input whose weight coefficients generally differ for every neuron. These weights $\theta := (W^1, b^1, \dots, W^4, b^4)$ are obtained by training the neural network on the available data set $\mathcal{T}$, which amounts to approximately minimizing a loss function

$$\mathcal{L}(\theta) := \frac{1}{|\mathcal{T}|}\sum_{(i,j)\in\mathcal{T}} f\big(d_{ij}, y_{\mathrm{NN}}^{(i,j)}\big);$$

we used a mix of mean squared and mean absolute percentage error: $f(x, y) = \frac{1}{\mu}(x - y)^2 + 100 \cdot |\frac{x-y}{x}|$, where $\mu := 3 \cdot 10^{-8}$ was chosen such that the two terms were about equally important.

For details on the implementation of network initialization and training method (including learning rate, dropout, etc.), we refer to Section 4.1.

2.3. Poisson pseudo-maximum-likelihood estimation

As mentioned in Section 1.1, Santos Silva and Tenreyro (2006) showed that using OLS to estimate the gravity model parameters via the log-transformed equations (2) is statistically inappropriate except under very specific assumptions which are unlikely to hold in practice. Briefly, Jensen's inequality—the expected value of the logarithm of a random variable is different from the logarithm of its expected value—implies that the logarithmized variables obtained by OLS may yield biased (inconsistent) estimates and misleading results when applying the original-scale model (1), unless the residual errors follow a normal distribution or have homogeneous variance; for full details, see Santos Silva and Tenreyro (2006), Motta (2019) and Santos Silva and Tenreyro (2022). This issue can be mitigated by using *Poisson regression with robust standard errors*, or more specifically, the (consistent) *Poisson pseudo-maximum-likelihood (PPML)* estimator. To that end, one can rewrite the original gravity formula (1), where we omitted the constant term β_0 for convenience, as

$$d_{ij} = e^{\sum_{k=1}^{K} \log\left(P_k(i,j)\right)\beta_k},$$

and then compute the model coefficients $\hat{\beta}_k$ by solving the first-order condition system

$$\sum_{(i,j)\in\mathcal{T}} \left(d_{ij} - e^{\sum_{k=1}^{K} \log\left(P_k(i,j)\right)\beta_k} \right) \begin{pmatrix} \log(P_1(i,j)) \\ \vdots \\ \log(P_K(i,j)) \end{pmatrix} = 0, \tag{9}$$

where the sum is over all data points, corresponding to city-pairs (i, j), available for model calibration. In short, system (9) is typically solved by an iterative second-order (Newton-like) method, and to at least partially account for heteroskedasticity, a robust covariance estimator (i.e., Hesse matrix approximation) is used for inference, see Santos Silva and Tenreyro (2006) and Motta (2019) for detailed derivations and further background.

3. Data

The data for our estimation models was collected in several meticulous steps. In the following, we first describe what we call *base data*; this encompasses, e.g., the list of European airports that serves as the basis for data collection and predicted demand evaluation, and some attraction parameter values that can be derived directly from that list. Subsequently, we will provide details on how we obtained the values of the remaining attraction parameter values. These data stem from public databases that we propose to utilize now and for future updates of the data set for air passenger demand estimation. To avoid possible data-induced prediction biases, the data on which to train the models should ideally stem from highly comparable sources, or even a single source. Thus, for the majority of data points, we employ the quite comprehensive, reliable and regularly updated Eurostat database (Eurostat, 2022), filling in a few missing values from other official and publicly accessible sources. To enable users to directly employ our data and code, and reproduce our experiments, we make our base data available online along with the program source files to gather and process external data (Joormann et al., 2023). Note that our published code also includes a script to automatically download the required external data files as well as a very detailed description of every single such file. (As mentioned earlier, practitioners may have access to additional or "better" data that is proprietary and only purchasable at relatively steep prices, if at all—whenever this is the case, its use is certainly encouraged; here, we must make do without.)

3.1. Base data

With the public availability of data (at a NUTS-3 or analogous level of resolution) in mind, we first selected a total of 36 European countries to represent what we consider the European air traffic network here, namely the 27 countries of the European Union (Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Poland, Portugal, Romania, Slovenia, Slovakia, Spain, Sweden), the United Kingdom, the EFTA countries Iceland, Norway, and Switzerland, as well as the EU candidate and potential candidate countries Albania, Montenegro, North Macedonia, Serbia and Turkey. Overseas territories and autonomous regions such as Greenland or the Channel islands are not included due to a lack of data, but several island groups like the Azores or Canary Islands are.

Recall that several of the attraction parameters we employ have been defined based on the EU statistical unit classification system NUTS.[1] Much of the Eurostat data is available at different resolutions with respect to statistical regions into which the EU and candidate / EFTA countries are subdivided. In particular, the NUTS-1 level contains whole countries as regions, the NUTS-2 level contains federal states or similar regions within each country, and the NUTS-3 level comprises the smallest statistical regions (e.g., in Germany, the different counties/rural districts called "Landkreise"). Moreover, Eurostat provides definitions for metropolitan areas that can cover multiple NUTS-3 regions.

We compiled a list of the 853 busiest airports in Europe (based on 2018 flight movements), along with the associated pairwise great-circle distances. After removing airports not located in any of the countries or territories under consideration as well as oil rigs and non-civil airports, 748 airports remained, for which we then identified the NUTS-3 regions they are located in. There are several metro areas or NUTS-3 regions that contain two or more of these airports. Attraction parameters such as population or GDP would hence be identical for such airports, regardless of their size or relevance. Therefore, we aggregated such airports and treat them as one. More precisely, we aggregated airports that are located in the same lowest-level statistical region (NUTS-3) or metropolitan area, which can encompass multiple NUTS-3 regions, and thus arrived at a final total of 531 "pseudo-airports"—meaning both aggregated and single airports—to work with. The distances between locations are averaged over the respective distances for all involved airports of each multi-airport city/region. We did not introduce an additional model variable that indicates or characterizes multi-airport locations, as done in, e.g., Grosche et al. (2007) and Rengaraju and Thamizh Arasan (1992), as this would amount to an air service related parameter, which we generally avoided, cf. Section 1.2.

The above-described choice of countries and pseudo-airports now builds the backbone of our proposed data set and for our methodology to gather the remaining data from (mostly) the Eurostat database. The base data we provide thus consists of:

- An indexed list of the 531 pseudo-airports with their respective (lists of) ICAO airport codes, and city/airport and country names.
- For each pseudo-airport, lists of the associated NUTS-3, metropolitan area (if applicable) and NUTS-2 identification codes, as well as a binary flag to distinguish whether a pseudo-airport lies in a Euro-zone country (1) or not (0); NUTS-3 codes are given for each actual airport location as well as, where applicable, for the whole respective metro area.
- The great-circle distance in kilometers between all pseudo-airport pairs, averaged with respect to distances between all involved airports in case of aggregation.
- After the Brexit, the United Kingdom is no longer a part of the EU and Eurostat apparently no longer gathers all statistical data for the UK's NUTS regions. Nevertheless, it seems the UK has simply relabeled the NUTS-3 regions as "ITL3" regions and is now reporting data for those, published by the UK's Office for National Statistics. We provide a list mapping ITL3 region codes to NUTS-3 classification codes to enable consolidating databases appropriately. Similarly, a mapping from Swiss cantons to NUTS-3 codes is provided as well.
- A list indicating for each NUTS-3 region whether it is "coastal", i.e., whether it has an ocean coast.
- Finally, we provide a selective mapping from the older NUTS-2016 classification scheme to the current NUTS-2021 scheme, which is needed to extract some data only available under the old classification codes; see our code and documentation for more details.

Note that while we only consider airport locations in this paper, the above kind of data can just as well be defined for arbitrary cities, whether they have an airport or not, and thus be used to estimate air travel demand involving origin and/or destination cities disconnected from the existing flight route network.

[1] We use the NUTS-2021 classification scheme; some data provided by Eurostat has not yet been adapted to changes in the NUTS classifications, so we manually map older NUTS-2016 codes to the current system whenever possible without ambiguity. See our code documentation for the full details.

3.2. Externally sourced statistical demographic and geo-economic data

Data for the attraction parameters was obtained almost exclusively from Eurostat via its publicly accessible online database (https://ec.europa.eu/eurostat/data/database). To avoid distorting effects caused by the ongoing CoViD-19 pandemic, we used data from 2019 whenever possible. (Recall that the models considered here take annual data from one year as input to produce estimates for the same year; forecasts for future years can be obtained by using forecasted values of the parameters.) Occasionally, single data entries were missing from the Eurostat files, in which case we used the corresponding latest available (pre-2019) data point instead. Moreover, we filled systemic gaps in the Eurostat database from other official and reliable sources; more precisely, GDP data for Switzerland and the United Kingdom was obtained from the Swiss Bureau of Statistics (https://www.bfs.admin.ch) and the UK's Office for National Statistics (https://www.ons.gov.uk), respectively. Also, as these values are provided in CHF and GBP, respectively, we use the average currency exchange rates for 2019 to convert them to EUR, the currency the remaining GDP data is reported in.

The different attraction parameters and their derivation from Eurostat/NUTS-based data have already been described in Section 2.1. As a proxy for the actual, but not publicly available, air passenger volume between two given cities i and j, we resort to taking d_{ij} as the passenger volume on existing non-stop flight connections between (pseudo-)airports in cities i and j. We again use Eurostat data, namely the numbers of "passengers on board" reported.

Our Python code (Joormann et al., 2023) downloads and extracts the necessary data from the Eurostat and statistics offices' websites and, based on our base data, performs the computations to compile the attraction parameter data in the assumed form (see above and Section 2.1.1), and creates a log file informing about missing or pre-2019 data points encountered. Rather than listing here the names of the various separate data files obtained from said sources, we refer to our code and, in particular, the data source documentation file, to check which files contained which information. The data we used was downloaded from the respective sites on 11/25/2022; our code should also work with future updated data files from the same sources, or be easily adapted to possible changes in their structure, e.g., once the transition to NUTS-2021 classifications will be completed.

We ended up with 3981 complete data points for the model calibration, i.e., values for d_{ij}, $P_1(i,j)$, $P_2(i,j)^+$, $P_2(i,j)^*$, …, $P_7(i,j)^+$, $P_7(i,j)^*$, $P_8(i,j)$, …, $P_{13}(i,j)$ for 3981 pairs (i,j) that correspond to (pseudo-) airport-pairs. In total, 308 of the 531 pseudo-airports occur in this data set, and 33 of the 36 countries are involved. The lower number of pseudo-airports is largely due to a lack of passengers-on-board data, eliminating 208; the remaining 15 are excluded because the appropriate GDP and/or tourism-nights data are missing or currently inaccessible due to the incomplete transition from NUTS-2016 to NUTS-2021 and associated border changes of some NUTS regions. As a result, no airports in Iceland, Albania or Serbia remained.

4. Computational results

We will first give a few details on our implementation and evaluation process, and then discuss the results of our computational experiments with the different approaches to air passenger demand estimation.

4.1. Implementation details and evaluation approach

We implemented the different methods to predict air passenger demand (cf. Section 2) in Python 3. The cardinality-constrained least-squares problems (4) were solved using the Python API of Gurobi 9.5.1 (Gurobi Optimization, LLC, 2022) in their reformulation as mixed-integer quadratic programs of the form (5). Although a suitable value of Big-M constants is generally not known a priori, in our experiments, the relatively small value $M = 20$ was sufficient and did not cause any numerical instabilities in the solving processes. (All of Gurobi's solver parameters were left at their respective default values except the MIP gap tolerance, which was set to 10^{-9}.) Furthermore, we employed statsmodels 0.13.2 (Seabold and Perktold, 2010) for PPML and some statistical output for OLS, and the machine learning libraries scikit-learn 1.0.2 (Pedregosa et al., 2011) and TensorFlow 2.8.0 (Abadi et al., 2015) with Keras 2.8.0 (Chollet et al., 2015); see our code and documentation for details. For the neural network model, each numerical feature was preprocessed before training to have zero mean and unit standard deviation using the standard Z-score normalization; the categorical parameters were supplied as binary values, without normalization. The neural network weights for the ReLU layers were initialized with the robust uniform method proposed by He et al. (2015) and those of the tanh-layer with the Glorot uniform initializer (Glorot and Bengio, 2010). The dropout regularization randomly removes neuron connections between two successive layers with fixed probability of 0.05 and 0.01 for the layer pairs $(1,2)$ and $(3,4)$, respectively. The training was performed with the Keras implementation of the Adam algorithm (Kingma and Ba, 2014) with batch size 64 and learning rate 0.002 for at most 10 000 epochs, terminating early if progress in terms of the validation loss (when using cross-validation) stalled over 300 epochs and restoring the corresponding best weights. Indeed, a stalling-based termination criterion for neural network training is sensible, since otherwise, the model may tend to overfit by adapting too much to the training data. The typical training behavior is shown in Fig. 2, where we plot the range (shaded areas) and average of training and validation error progression on an exemplary random 10-fold data split. Letting the respective training proceed the full 10 000 epochs achieves no further improvement in predictive power (measured in terms of the validation loss), yet keeps reducing the training loss until final goodness-of-fit values of, e.g., $R^2 > 0.9$, which may seem great but is, in fact, a clear sign of overfitting since the performance on withheld (validation) data stagnates.

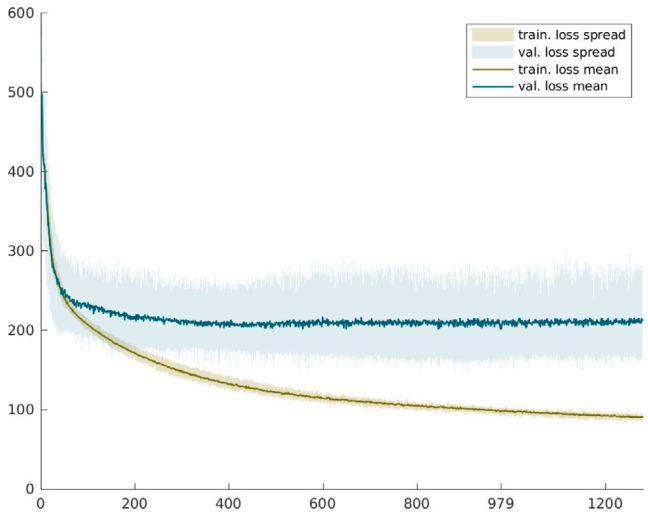

Fig. 2. Exemplary progression of training and validation loss (means and spreads) in the course of neural network training on a random 10-fold data split. Terminating early soon after validation loss first stops improving prevents model overfitting (training losses continue to drop over 10 000 epochs while validation losses stagnate after, on average, 979 epochs).

Let $N := 3981$ denote the number of data points, and n the number of model variables, which equals $K := 19$ if all attraction parameters are used, but may be lower in models incorporating a feature selection mechanism. Furthermore, let y^k and $\hat{y}^k$ be the given and predicted values for the air passenger volumes for data points indexed by $k \in$

$[N] := \{1, 2, \dots, N\}$, respectively; i.e., each y^k corresponds to d_{ij} for some city-pair $(i, j) \in \mathcal{T}$, and analogously for $\hat{y}^k$.

We utilize several performance measures to provide a more complete picture of the behavior of the different estimation models: We compute the *coefficient of determination*, also known as R^2 *statistic*, see, e.g., James et al. (2021), i.e.,

$$R^2 := 1 - \frac{\sum_{k=1}^{N}(y^k - \hat{y}^k)^2}{\sum_{k=1}^{N}(y^k - \frac{1}{N}\sum_{k=1}^{N} y^k)^2},$$

as well as the *mean absolute error (MAE)*

$$\text{MAE} := \frac{1}{N}\sum_{k=1}^{N}|y^k - \hat{y}^k|,$$

the *maximal loss* or *maximal absolute deviation (MAD)*

$$\text{MAD} := \max_{k\in[N]}|y^k - \hat{y}^k|,$$

and the *mean absolute percentage error (MAPE)*

$$\text{MAPE} := \frac{1}{N}\sum_{k=1}^{N}\frac{|y^k - \hat{y}^k|}{|y^k|}.$$

For R^2, which is upper-bounded by 1, larger values are generally better. For all other scores, it holds that the smaller their value, the better. We favored MAE over the (root) mean squared error, another common metric, since MAE is more robust with respect to possible large outliers. Moreover, we will assess the above metrics using cross-validation to judge estimation performance on unseen data (cf. also Shmueli, 2010).

Note that these measure are all defined with respect to the *original* data and not their logarithmized form, since we want to assess the *actual* prediction errors after model calibration. It seems that R^2 values commonly found in the related literature always pertain to the log–log system (2), and thus specify the goodness of the OLS regression but, in fact, not directly the resulting prediction accuracy of the respective underlying gravity model (1). Furthermore, as mentioned earlier, R^2 has been criticized as being easily misinterpreted or misleading, see, e.g., Birnbaum (1973), Berk (2004) and Anderson and Shanteau (1977), and therefore should not be relied on solely. Nevertheless, for completeness and some degree of comparability with existing works, we will also give the R^2 values with respect to the logarithmized model for all methods that are based on it. Moreover, let us remark that the widespread interpretation of R^2 as the portion of variance explained is specific to linear regression; for other models and approaches, various variants of R^2 are commonly used, e.g., McFadden's pseudo-R^2 (McFadden, 1974) for PPML/Poisson regression. Since such variants are not directly comparable and the standard R^2 can still serve as a measure for how well estimations match known data, we refrain from including further different R^2 scores.

4.2. Experimental results and discussion

For clarity, we break down the discussion of computational results into three parts. Firstly, in Section 4.2.1, we calibrate the "full" models using all of the features defined earlier. We evaluate the various methods based on repeated 10-fold cross-validation (CV) for the sake of robustness; see, e.g., James et al. (2021). To that end, we randomly split the data set into 10 subsets of about equal size (the *folds*), train the models 10 times, each time withholding a different fold for validation and using the remaining 9 folds as training data. To further mitigate any biases induced by a concrete data split, we repeat the 10-fold CV 100 times and average all results to assess goodness-of-fit and how the trained models perform on unknown data. For further comparison, we also show the results obtained from model calibration taking *all* of the available input data into account; in that case, the scores only represent goodness of fit to the given data, as no evaluation on data not used for training is possible, and especially the more advanced models may tend to overfit.

In Section 4.2.2, we address the question of feature selection. To that end, we not only consider selection based on the common notion of statistical significance, but, for all possible numbers of features, we compare the results of calibrating the gravity model using a cardinality-constrained least-squares approach as well as two stepwise regression methods, based on 100 repetitions with 10-fold CV-like data splits. We discuss the identification of a sensible number and choice of features and, for all three approaches, provide the prediction quality (CV-) scores of all reduced models considering only the respective feature subsets.

Finally, in Section 4.2.3, we investigate the effects of allowing only one of each pair of the intrinsically related "sum- and product-features" $(P_k^+(i,j), P_k^*(i,j),\ k = 2, \dots, 7)$ to enter the model. In principle, there is no prohibitive factual reason against including both such features. However, it may be argued that this could introduce unwanted correlation into the model, that not both of a pair could be statistically significant, or that it would at least constitute a questionable feature design not to decide on one form, sum or product, to begin with. We will not discuss correlation or (multi-) collinearity aspects since these can have varying meaning for the different models and because dropping one of a pair of correlated variables can, in turn, lead to biased results via absorption effects, cf. Jorge-Calderón (1997). Thus, we will inspect feature significance and, in particular, compute the optimal combination of feature designs, exclusively allowing either the sum or product feature of each respective pair, by means of mixed-integer quadratic programming. We then again assess the cross-validated accuracy of all considered estimation models when restricted to the selected optimal features.

We remark that, while we will not report coefficient values resulting from any of the regression methods or the neural network model, our results—including the calibrated models with the respective coefficients—can be reproduced by the code that can be obtained from our online repository (Joormann et al., 2023), up to small variations in CV scores or neural network training that are due to randomization (exact replication is possible by using our provided script and identical package versions, cf. the code documentation).

4.2.1. Full model calibration and cross-validated accuracy assessment

Tables 2 and 3 show the performance measure values (or scores, for short) for all methods for the 100× repeated 10-fold cross-validation experiments and the models calibrated on all available data, respectively. In both tables, the columns are labeled according to the different approaches, ordinary least-squares regression (OLS), least absolute value regression (LAV), Poisson regression with robust standard errors (Poisson pseudo-maximum likelihood, PPML), kernel ridge regression (KRR), support vector regression (SVR) and neural network (NN). Note that Table 2 gives average scores over all cross-validation runs whereas calibration with all data (Table 3) requires only one run, though to account for randomization in the neural network initialization and training algorithm, NN results are again averaged over 100 runs. For additional comparison, we also evaluate a "direct OLS" approach that fits a linear regression function with respect to the original, non-logarithmized data and features, i.e., considers (2) but without taking logarithms. The reported scores in all subsequent tables are all rounded to four significant digits except MAE and MAD, which were rounded to the nearest integer.

Let us first look at the results provided in Table 2. Firstly, from comparing the average scores on just the training and just the validation data sets, respectively, we can conclude that none of the models suffer from strong overfitting effects. This would be recognizable by dramatically better results on the training data than on the validation set (which was withheld during model calibration), but while such discrepancies can indeed be observed for each model, especially regarding R^2 scores, the respective value pairs are still reasonably similar for all different scores. Unsurprisingly, the largest differences appear for the neural network, and while not worrisome (validation losses stagnate

Table 2
Results of 100× repeated 10-fold CV using all features ($n = K$). The first row block gives the average CV scores, the second and third block the average training and validation scores, respectively.

Measure	Direct OLS	OLS	LAV	PPML	KRR	SVR	NN
R^2	0.2955	0.2386	0.2444	0.4355	0.3818	0.4605	0.6831
R^2 (log.)	–	0.3838	0.3796	–	0.4712	0.5116	–
MAE	135974	116333	115779	122098	104223	98684	83300
MAD	2834897	2892485	2879965	2348201	3046258	2626100	1830872
MAPE	1.6142	0.8380	0.8061	1.2776	0.7174	0.6794	0.5103
R^2 (tr.)	0.2985	0.2390	0.2447	0.4370	0.3857	0.4644	0.7051
R^2 (log.) (tr.)	–	0.3845	0.3803	–	0.4783	0.5161	–
MAE (tr.)	135859	116287	115710	121984	103859	98203	80991
MAD (tr.)	2776908	2885770	2875355	2333545	3028083	2604792	1611814
MAPE (tr.)	1.6135	0.8373	0.8053	1.2768	0.7135	0.6746	0.4942
R^2 (val.)	0.2595	0.2352	0.2403	0.4105	0.3488	0.4231	0.4772
R^2 (log.) (val.)	–	0.3750	0.3705	–	0.4040	0.4688	–
MAE (val.)	137007	116747	116400	123122	107505	103016	104085
MAD (val.)	2189417	2235202	2221620	1727728	2232077	1994553	1626807
MAPE (val.)	1.6207	0.8440	0.8133	1.2852	0.7518	0.7221	0.6550

Table 3
Results for model calibration using all features ($n = K$) on full data set.

Measure	Direct OLS	OLS	LAV	PPML	KRR	SVR	NN
R^2	0.2953	0.2386	0.2431	0.4361	0.3945	0.4671	0.9753
R^2 (log.)	–	0.3841	0.3804	–	0.4836	0.5177	–
MAE	136320	116311	115773	122034	103226	98070	24673
MAD	2782686	2892653	2891030	2340407	2995454	2588182	614064
MAPE	1.6252	0.8376	0.8039	1.2772	0.7109	0.6748	0.1753

but do not keep increasing much, cf. Fig. 2), they indicate there should indeed be room for further improving[2] the network architecture. Here, the neural network training was stopped after 731 epochs on average across all CV experiments.

Turning to the overall cross-validation scores, several aspects stand out. For one thing, contrary to what one might expect, OLS regression based on the log–log equations (2) resulting from the gravity model (1) is not obviously preferable to using a simple linear estimation function: Direct OLS achieves R^2 values that are notably better than those of OLS, whereas the MAE and MAPE scores are much worse; MAD values are comparable. Nevertheless, the coefficient of determination for OLS with respect to logarithmized data shows that the log–log equations admit a better linear estimation than the original parameters, as indicated by the R^2 of 0.3838 for the former versus 0.2955 for the latter. This can be seen as further evidence of the general validity of the gravity model approach as opposed to a fully linear model. In fact, given that the advantage of direct OLS over OLS with respect to the MAD score is comparatively small but the accuracy loss indicated by MAE and MAPE quite large, the gravity model OLS ultimately does seem more suitable than the direct OLS approach despite the contraindication in terms of R^2 scores on the non-logarithmized data.

However, regarding the calibration method for a gravity model, the other results show that the ubiquitous OLS approach is, in fact, clearly inferior to all other considered methods. While LAV may seem slightly worse than OLS if one were to focus solely on the R^2 values that pertain to the log–log equation system (2)—as is typically done in previous studies of gravity models for air passenger demand estimation—looking at any one of the four other performance measures, i.e., those that are based on the actual (non-logarithmized) demand/volume numbers of interest, one can see that LAV actually yields slightly *better* results than OLS. Moreover, moving further to incorporating nonlinear kernels as in KRR, employing a (kernelized) SVR and finally a neural network model, we can see the scores become increasingly better. The improvement over OLS achieved by KRR, by and large due to kernelization as the major difference between the two approaches, is already quite impressive, and can be pushed even further by replacing the underlying least-squares approach of OLS and KRR by the support vector machine employed in SVR, and ultimately, by dropping the gravity model itself and instead utilizing a neural network. (We reiterate that the neural network here was not constructed to be particularly well-suited for our purposes and that further improvements are very likely achievable by conducting a thorough neural architecture search; similarly, using different kernels in KRR and SVR could also lead to better respective performance.)

Recall that the PPML estimator had been proposed as an alternative to OLS with respect to (2) due to preferable statistical properties from the viewpoint of expected values, error distributions and heteroskedasticity issues, cf. Santos Silva and Tenreyro (2006). Indeed, we can observe much better R^2 and MAD scores for PPML than OLS, and also LAV and even KRR. Also, the improvements over direct OLS clearly demonstrate that the nonlinear multiplicative model (1) is more suitable than a simple linear model. However, the MAE and MAPE scores of PPML are the second worst among all seven approaches, outperforming only the fully linear regression. Together with the comparatively good MAD score, this indicates that PPML yields an estimation function that achieves a low largest deviation at the price of much larger average deviations. In other words, PPML gives a smaller upper bound on the absolute errors, but these errors are larger on average than for the other methods, except direct OLS. Conversely, the other approaches (again, except direct OLS) apparently reduce the average individual deviations significantly further, but accept occasional large absolute errors in return. Thus, whether PPML is indeed preferable to OLS, or the LAV and KRR approaches based on the log-linearized gravity model, depends on which error distribution behavior is more acceptable to the user. In fact, such a choice may not be needed when employing the final two methods, SVR and NN: SVR gives notable improvements with respect to all estimation quality measures except MAD, and the neural network very clearly outperforms across *all* performance scores. Even focusing only on the validation scores, which here provide the best idea of calibrated model prediction behavior on unseen data alone, the neural network outperforms by a notable margin, with the single exception of having a slightly larger MAE score than SVR.

Moving on to Table 3, we see that, as expected, the individual results become a bit better than the cross-validated scores when all

[2] It is also interesting that all methods except NN exhibit notably lower average MAD scores on the validation sets than on the training data, though all other scores are slightly worse.

available data is employed to calibrate the different models—though training the NN for 10 000 epochs leads to overly adapting to the known data (overfitting), so for the NN, we should certainly focus on the CV or validation scores reported in Table 2—but the conclusions remain the same. Using all available data for model calibration/training should be avoided unless one can be reasonably certain that no problematic overfitting occurs; for the neural network architecture used here, 10 000 epochs is clearly too much, and should be replaced by a number around, say, 1 000 if one wants to train with the full data set. In all cases, we recommend relying on cross-validated performance, and view the results from Table 3 as complementary information only.

All in all, the results reported in Table 2 (and Table 3) show that, in particular, the machine learning techniques (KRR, SVR and NN) yield significantly better scores than the simpler OLS or LAV approaches, as does PPML, which is nevertheless inferior to SVR and especially NN. The overall best model turns out to be the neural network, which yields average CV-score improvements between 28.4 % and 186.3 % in all performance measures compared to the standard OLS technique, as well as 22.0 % to 60.1 % and 15.6 % to 48.3 % compared to PPML and SVR, respectively (cf. Table 2).

To further support our conclusion that the neural network model outperforms all other approaches, we compared the means of the city-pair-wise differences between the ground-truth and model estimation results for the neural network against the other methods. Since the performance scores reported in Tables 2 and 3 are average values over all city-pairs, for each model separately, it *might* happen that some such average indicates that one model performs better than another even though the latter actually gives results closer to the given demand numbers for the majority of connections, but has a few large outliers. Therefore, we tested whether the neural network predictions are indeed significantly better (in a statistical sense) than those of the other models in terms of route-based estimation quality. To that end, we conducted left-tailed paired t-tests (see, e.g., Lehmann and Romano, 2005; Kanji, 2006) of route-by-route differences for the neural network versus each other approach, which confirmed the superiority of the neural network model at an overall significance level of well below 0.000 1 %; see Appendix A for the details.

Finally, regarding a comparison with existing related models, we again emphasize that our results cannot be compared directly to other scores reported in the literature due to differences in feature definitions, model specifications and, in particular, the underlying data set. Nevertheless, it is worth noting that the "R^2 (log.)" values we achieve on our publicly available data do appear to be competitive with existing results that depended on proprietary data. For instance, the R^2 value 0.5177 of the SVR model with respect to the log–log equation system (2) (from Table 3; similarly for the corresponding validation score 0.4688 from Table 2) compares favorably with the score of 0.283 reported for a related earlier gravity model for Europe in Grosche (2009) or the "reasonably high" value 0.47 of a model that had been used in Canada (International Civic Aviation Organization, 2006), yet falls short of occasional very high values such as the R^2 of 0.952 for the gravity model for India from Rengaraju and Thamizh Arasan (1992). Moreover, such very high values were typically achieved only on very small data sets and more specific regions than the relatively broad European market we consider here, and might even indicate overfitting (cf. the scores for the neural network model in Table 3). In contrast, our data set is significantly larger than most others that had been employed for air passenger demand estimation, cf., e.g., Grosche et al. (2007, Table 1), and offers enough data points for meaningful cross-validation to identify possible overfitting issues. Depending on the application, it may be adequate to further divide the European market into smaller, more homogeneous segments, e.g., by country, and train separate models for the individual segments (possibly including additional region-specific variables) to achieve further improved performance; such specialization would go beyond the scope of the present paper and is thus left for future work.

4.2.2. Feature selection and reduced model evaluation

We now turn to feature selection in order to either eliminate (statistically) insignificant variables or reduce model complexity by removing parameters that yield no relevant further accuracy improvement when included. For brevity, we will henceforth refer to the features mostly by their mathematical variable names; see Table 1 to look up the corresponding definitions. In the context of gravity models and air passenger volume estimation, feature selection has been based on the standard OLS approach, and was either skipped entirely (when all originally included variables were deemed statistically significant) or performed with stepwise OLS regression as in, e.g., Rengaraju and Thamizh Arasan (1992). Therefore, we also focus on OLS and employ two stepwise schemes—sequential forward selection (SFS) (Ferri et al., 1994) and recursive feature elimination (RFE) (Guyon et al., 2002), see Section 2.1.3—as well as a modification of the cardinality-constrained version (5) of ordinary least-squares regression (CCLS for short) with "big-M" $M = 20$, all evaluated with cross-validation. The CCLS modification takes a CV-data split into account by minimizing the average training (least-squares) loss across all folds, but enforcing a combined decision on which features to use: Denoting by $\hat{P}^j$ and $\hat{d}^j$ the input data retained for training in the jth fold, for k-fold CV, we solve the *group-CCLS* problem

$$\begin{aligned} \min_{\{\beta^j\},y} \quad & \sum_{j=1}^{k} (\beta^j)^\top (\hat{P}^j)^\top \hat{P}^j \beta^j - 2(\hat{d}^j)^\top \hat{P}^j \beta^j \\ \text{s.t.} \quad & \mathbb{1}^\top y \le s, \quad -M y \le \beta^j \le M y \quad \forall j \in [k] \\ & \beta^j \in \mathbb{R}^K \quad \forall j \in [k], \quad y \in \{0,1\}^K; \end{aligned} \tag{10}$$

recall the notation $[k] = \{1, 2, \dots, k\}$. Note that the coefficients β^j can adapt to each fold j separately, but a single auxiliary binary vector is used to realize the cardinality constraint and ensure all β^j have the same support. Since the number of binary variables is the same as in the standard sparse regression program, (10) remains efficiently solvable (in the present context). Intuitively, enforcing that the same features are selected for all k folds that stem from one data split eliminates dependency on the fold used for training and should lead to more stable validation scores, i.e., behavior on unseen data. However, since already-trained models are needed for the evaluation of such CV scores, their explicit minimization would require the enumeration of all possible feature combinations and training of the resulting reduced models (on each fold)—this is essentially what RFE and SFS are based on, circumventing full enumeration by greedily selecting one feature at a time, with no mechanism to revise such decision in later iterations.

We remark that, in principle, feature selection could be performed based on any kind of method, including SVR or deep learning, though this could become very expensive computationally. Thus, while OLS-based decisions may not necessarily coincide with optimally selected features for other approaches, the hope is they can still be regarded as reasonable and efficiently obtainable options to reduce the feature count.

As a first step, one may also consider the so-called *p-values* of t-statistics; we refer to, e.g., Hastie et al. (2009), Murphy (2012) and James et al. (2021) for details on their computation and statistical background. It is common practice to decide on model parameters' statistical significance based on these p-values falling below a certain threshold α—typically, the 5 % or 1 % level, i.e., $\alpha = 0.05$ or $\alpha = 0.01$, respectively. However, similarly to R^2 scores, this approach has been criticized for often being misused, misinterpreted or even plain inadequate, see, e.g., Murphy (2012) and Colquhoun (2014). Hence, we will not dive deeply into this sort of statistical analysis here, but rather discuss the corresponding standard result interpretation vis-à-vis the outcomes of the aforementioned feature selection methods.

A well-informed feature selection often requires careful consideration of several decision criteria. Especially when utilizing cross-validation to assess results on a more robust basis, even "one-shot-methods" like RFE typically do not consistently yield the same outcomes on different folds, and the different performance scores do not

Table 4
Selection decision summary for feature selection (FS) based on 100× repeated 10-fold CV, average improvement stalling and variable selection frequencies (cf. Appendix B). The selection based on OLS p-value significance at the 5 % level is also reported.

FS method	s	Selected features
p-values (5 %)	12	P_2^*, P_4^*, P_4^+, P_5^+, P_6^*, P_6^+, P_7^*, P_9, P_{10}, P_{11}, P_{12}, P_{13}
group-CCLS	10	P_2^*, P_4^*, P_4^+, P_5^+, P_6^*, P_6^+, P_7^*, P_9, P_{10}, P_{12}
SFS	11	P_2^*, P_4^*, P_4^+, P_5^*, P_5^+, P_6^*, P_6^+, P_7^*, P_9, P_{10}, P_{12}
RFE	16	P_2^*, P_2^+, P_3^+, P_4^*, P_4^+, P_5^*, P_5^+, P_6^*, P_6^+, P_7^*, P_7^+, P_9, P_{10}, P_{11}, P_{12}, P_{13}

necessarily improve monotonically with the number of retained features. In order to keep the present discussion focused, we will provide a summary of our feature selection analysis in the following, and relegate the technical details and in-depth arguments that led to the specific feature subsets to Appendix B.

We summarize the features selected according to the different approaches in Table 4, where s stands for the corresponding numbers of chosen features. A few things stand out clearly from this overview: None of the FS methods includes the distance parameter P_1 or the same-currency indicator P_8. Also, the catchment parameters P_3^* and P_3^+ are almost always excluded except in the RFE results, and those for a coastal or island location, P_{11} and P_{13}, only occur in the p-values-based and RFE selection. The set of ten variables obtained by group-CCLS is also included in all other, larger feature sets. All methods produce feature subsets of different sizes, extending the group-CCLS selection by the PLI product P_5^* (SFS), the coastal and island location indicators P_{11} and P_{13} (p-values), or these three as well as the sum parameters for population (P_2^+), catchment (P_3^+) and poverty risk (P_7^+) in case of RFE. We also note that only group-CCLS produced a proper subset of the features that showed evidence of statistical significance at the 5 % level; nevertheless, p-values computed on the respective reduced models do not indicate insignificance among the features selected by SFS either, but for RFE, two features "fail" at the 5 % level and two more at the 1 % level.

To assess the suitability of these feature selections, we now evaluate all calibration methods based on 10× repeated 10-fold cross-validation, restricting the underlying models to the respective feature subsets. The resulting average CV performance scores are presented in Table 5; for brevity, we omit MAD scores, and since not all models employ the log-linearized form (2), we also leave out the R^2 scores that do not pertain directly to the original, non-logarithmized data.

Comparing the results in Table 5 to those in Table 2, we see that, as expected, reducing the number of features using any of the selection methods naturally decreases the accuracy of all models. Nevertheless, the decrease is fairly moderate even for the smaller feature subsets, and generally appears more pronounced with respect to the R^2 values than other scores. For direct OLS, the MAE and MAPE scores even improve when reducing the feature count; also, LAV with the p-value or RFE-based selection has a slightly better MAPE value than when using all features. The RFE selection removes only three features, so the associated scores are closest to those obtained using all features. However, Table 9 (in Appendix B) shows that even with as many as 16 features, group-CCLS—and also SFS—yields better results than RFE (for the standard OLS approach; note that in all experiments, improvements in OLS performance always implied improvements for the superior methods like SVR and NN as well). The results for group-CCLS-10, SFS-11 and p-12 are fairly comparable, with differences mostly limited to the third or second decimal. Notably, the machine learning approaches (KRR, SVR and NN) all improve in all scores when more features are included (except NN moving from 10 to 11 features), even though the respective selection procedure was different. On the other hand, for the other methods (direct OLS, OLS, LAV and PPML), the numbers are occasionally ambiguous; for instance, for LAV, SFS-11 yields the best R^2 score, but p-12 gives a better MAPE value. Nevertheless, again consulting Table 9, group-CCLS will outperform both the p-value-based selection (with 12 features) as well as SFS (with 11 features), at least for OLS.

Table 5
Results (average CV scores) for model calibration based on 10× repeated 10-fold CV, using selected features as in Table 4. Average NN training epochs: 609 (group-CCLS), 555 (SFS), 737 (p), 731 (RFE).

Method	Score	group-CCLS-10	SFS-11	p-12	RFE-16
direct OLS	R^2	0.2884	0.2859	0.2886	0.2952
	MAE	131864	133278	132950	134441
	MAPE	1.5197	1.5730	1.5646	1.5821
OLS	R^2	0.2281	0.2317	0.2309	0.2369
	MAE	116917	116719	116681	116348
	MAPE	0.8443	0.8440	0.8396	0.8384
LAV	R^2	0.2352	0.2445	0.2371	0.2409
	MAE	116292	115960	116088	115867
	MAPE	0.8110	0.8198	0.8030	0.8055
PPML	R^2	0.3938	0.4077	0.4047	0.4342
	MAE	124632	123781	124108	122076
	MAPE	1.2976	1.2927	1.2915	1.2789
KRR	R^2	0.3248	0.3277	0.3418	0.3561
	MAE	109255	108814	107340	106267
	MAPE	0.7487	0.7464	0.7335	0.7308
SVR	R^2	0.3931	0.3947	0.4140	0.4285
	MAE	104878	104776	103033	101462
	MAPE	0.7184	0.7182	0.7068	0.6950
NN	R^2	0.4669	0.4597	0.5412	0.5849
	MAE	105093	105705	98511	93120
	MAPE	0.6361	0.6416	0.5978	0.5592

Finally, we observe that the neural network model restricted to the group-CCLS-10 features still clearly outperforms all other approaches even when those were calibrated with *all* features (cf. Table 3); only MAE is slightly worse than that of KRR and SVR in this case, and R^2 negligibly so with respect to SVR. This again emphasizes the superiority of the proposed neural network over the traditional log-linearized OLS approach, PPML, and the other methods we covered. Moreover, omitting features from the model generally seems to more strongly affect the performance scores for NN than the other methods. This can be seen as evidence that only the neural network model can make significant use of additional features beyond a certain number, whereas the other methods have already reached their near-full potential with markedly fewer features than originally considered.

To summarize, while the feature selection schemes yielded somewhat comparable results, with small differences depending on the model and calibration approach, the group-CCLS approach consistently produced feature subsets that lead to overall better estimation accuracy scores than the selections by the other methods, and RFE appeared inferior to all others. It is also worth mentioning that group-CCLS was significantly faster than SFS, despite involving the solution of a mixed-integer quadratic program: The total runtime for all 100 repetitions on 10 folds for each $s \in \{1, \dots, 18\}$ was about 16 minutes for group-CCLS versus roughly 43 minutes for SFS (measured on a standard Linux machine with 16 GB memory and six Intel Core i7-8700 CPUs with 3.2 GHz), and group-CCLS was faster than SFS for all s individually as well. Thus, group-CCLS emerges as a fast and reliable method for consistent feature selection, both for identifying a smallest reasonable feature subset, and to provide an alternative selection that yields better performance of trained models if the number of features to keep is known a priori or was obtained from another

Table 6
Result summary for feature design selection via (11) based on 100× repeated 10-fold CV.

s	R^2	R^2 (log.)	MAE	MAPE	retained features
6	0.2064	0.3547	118848	0.8706	P_4^*, P_6^*, P_7^*, P_9, P_{10}, P_{12}
7	0.2134	0.3603	118096	0.8637	P_3^+, P_4^*, P_6^*, P_7^*, P_9, P_{10}, P_{12}
8	0.2159	0.3651	117704	0.8567	P_3^+, P_4^*, P_6^*, P_7^*, P_9, P_{10}, P_{11}, P_{12}
9	0.2177	0.3665	117669	0.8565	P_1, P_3^+, P_4^*, P_6^*, P_7^*, P_9, P_{10}, P_{11}, P_{12}
10	0.2228	0.3684	117235	0.8526	P_2^*, P_3^+, P_4^*, P_5^+, P_6^*, P_7^*, P_9, P_{10}, P_{11}, P_{12}
11	0.2251	0.3696	116990	0.8492	P_2^+, P_3^+, P_4^*, P_5^+, P_6^*, P_7^*, P_9, P_{10}, P_{11}, P_{12}, P_{13}
12	0.2283	0.3714	116895	0.8484	P_1, P_2^+, P_3^+, P_4^*, P_5^+, P_6^*, P_7^*, P_9, …, P_{13}
13	0.2291	0.3720	116836	0.8478	P_1, P_2^+, P_3^+, P_4^*, P_5^+, P_6^*, P_7^*, P_8, P_9, …, P_{13}

algorithm such as SFS. Moreover, in the following subsection, we will see that the (group-)CCLS approach—or, more precisely, the mixed-integer quadratic program—offers additional flexibility that can be used to automate certain feature design considerations.

4.2.3. Feature design: Effects of limiting inherently related sum/product features

Until now, a city-pair's population, catchment size, GDP, PLI, guest-nights and poverty-risk percentages all entered the demand estimation models via two respective features: their product and their sum. While, in principle, it is entirely up to the user how parameters are integrated in the model, it may be argued that multiple route-based features being based on the same city-based parameters could be an ill-conceived design decision due to, e.g., the introduction of undesired feature correlation or a blurring of model interpretability. Indeed, for instance, Rengaraju and Thamizh Arasan (1992) additionally considered the respective ratios and sum of reciprocal values besides products and sums, and decided on one form for each underlying parameter based on correlation arguments. Since having highly correlated features may affect different models and algorithms in different ways, a rigorous statistical analysis of these aspects for all models compared here would go beyond the scope of this paper. Instead, as for the exemplary feature selection discussion above, we focus on the log–log gravity model (2) and equip the standard OLS approach with a means to automatically include the best combination of feature designs for the respective parameters. To that end, we extend the group-CCLS model (10) and solve the mixed-integer quadratic program

$$
\begin{aligned}
\min \quad & \sum_{j=1}^{k} \tfrac{1}{2} \|\hat{P}^j \beta^j - \hat{d}^j\|_2^2 && && (11)\\
\text{s.t.} \quad & -M y \le \beta^j \le M y && \forall j \in [k] \\
& \mathbb{1}^\top y \le s \\
& y_{2\ell} + y_{2\ell+1} \le 1 && \forall \ell \in \{1, 2, \dots, 6\} \\
& \beta^j \in \mathbb{R}^K \quad \forall j \in [k], \; y \in \{0,1\}^K,
\end{aligned}
$$

where $\hat{P}^j$ and $\hat{d}^j$ again denote the input data in the jth fold, $j = 1, \dots, k$. As in the sparse regression problems (5) and (10), the big-M constraints force all β_k^j to 0 if $y_k = 0$, and $\mathbb{1}^\top y \le s$ thus limits the allowed number of features to s and ensures the β^j, $j \in [k]$, share a common support. Additionally, we now force the solver to decide between the product- and sum-based features with so-called *SOS1-constraints*: each constraint $y_{2\ell} + y_{2\ell+1} \le 1$ allows exactly one of $\beta_{2\ell}^j$ and $\beta_{2\ell+1}^j$ to be nonzero and, consequently, admits only one of the corresponding features $P_{\ell+1}^*$ and $P_{\ell+1}^+$ to be included in the model calibrated by solving (11). In our experiments, a big-M constant $M = 20$ was again sufficiently large and did not cause numerical problems for the MIP solver Gurobi.

The averaged results of 100× repeated 10-fold CV-based experiments are summarized in Table 6; since for $s \le 5$, the group-CCLS model (10) already never chose both of any pair of related features, and because six of the 19 features will always be excluded by the new constraints $y_{2\ell} + y_{2\ell+1} \le 1$, we only report the results for $s \in \{6, 7, \dots, 13\}$. Results for $s \le 5$ correspond to those reported for group-CCLS in Table 9, and those for all $s \ge 14$ are the same as those for $s = 13$. For brevity, we again omit MAD values, but state which ones of the features in question were included in the model; the respective selection frequencies among the repetitions consistently were 100 % in all cases.

Table 7
Results (average CV scores) for reduced models based on 10× repeated 10-fold cross-validation, using features selected by (11) with $s = 13$. Average NN training epochs: 725.

Measure	direct OLS	OLS	LAV	PPML	KRR	SVR	NN
R^2	0.2942	0.2290	0.2439	0.4174	0.3698	0.4453	0.5874
MAE	135248	116852	116026	123383	105819	100627	93267
MAPE	1.5863	0.8481	0.8213	1.2999	0.7391	0.7009	0.5662

From Table 6, we can see that across the considered feature number range, the (OLS) model quality remains comparable, but always slightly below what can be achieved when allowing feature-pairs (except regarding R^2 for $s \in \{6, 7\}$, cf. Table 9). Notably, in most cases, it seems very clear which of the two feature designs is preferable according to the modified CCLS model (11): P_3^*, P_4^+, P_5^*, P_6^+ and P_7^+ indeed never occur in any of the selections for any s, and P_2^* is chosen only for $s \in \{1, 2, 10\}$ (see Fig. 3 for $s \le 5$), with P_2^+ being preferred for $s \in \{11, 12, 13\}$. In-line with previous observations from Section 4.2.2, the same-currency indicator P_8 appears to be the least important, only being added for the final $s = 13$. However, in contrast to the earlier feature selection experiments, the SOS1-constraints lead to increased relevance of distance (P_1) and catchment (P_3^+) in particular.

Looking back at Table 4, we can also observe that without the SOS1-constraints, every feature selection method included at least one pair of related features, even group-CCLS with just 10 total features. Thus, the selections suggested by (11) are genuinely different yet again from the previously considered feature subsets. For a brief comparison, Table 7 (analogously to Tables 2 and 5) summarizes the cross-validated performance scores for the full model under SOS1-constraints, i.e., utilizing $s = 13$ features but only one of each pair of related ones; per Table 6, this means excluding P_2^*, P_3^*, P_4^+, P_5^*, P_6^+ and P_7^+. We note that the reduced-model OLS-based p-values for this selection indicate that all retained features are statistically significant at the 1 % level.

The present selection of features turns out to yield results that are quite comparable to those achievable when admitting all features—of course, there is some unavoidable overall loss in accuracy for all methods, but the decrease with respect to the corresponding results in Table 2 is rather small, also compared to the feature subsets evaluated in Table 5. In particular, the neural network again stands out as the clearly best model across all scores. Thus, we can conclude that in case one is concerned about interpretability, indecisive feature design or possible correlation issues, it does not do much harm to allow only one of each "feature pair" such as the sum and product ones we considered to enter the model. Moreover, the selection of the most suitable designs does not need to be done manually, but can be performed automatically with the proposed program (11) with SOS1-constraints. While designed based on the log-linearized OLS approach, the resulting selection still gives good results for the other models and calibration methods, including the neural network. Also, (11) naturally incorporates a cardinality-constraint which can be used for further feature selection—as seen in the previous subsection, the flexible group-CCLS approach is competitive with or even superior to other feature selection schemes (cf. also Appendix B).

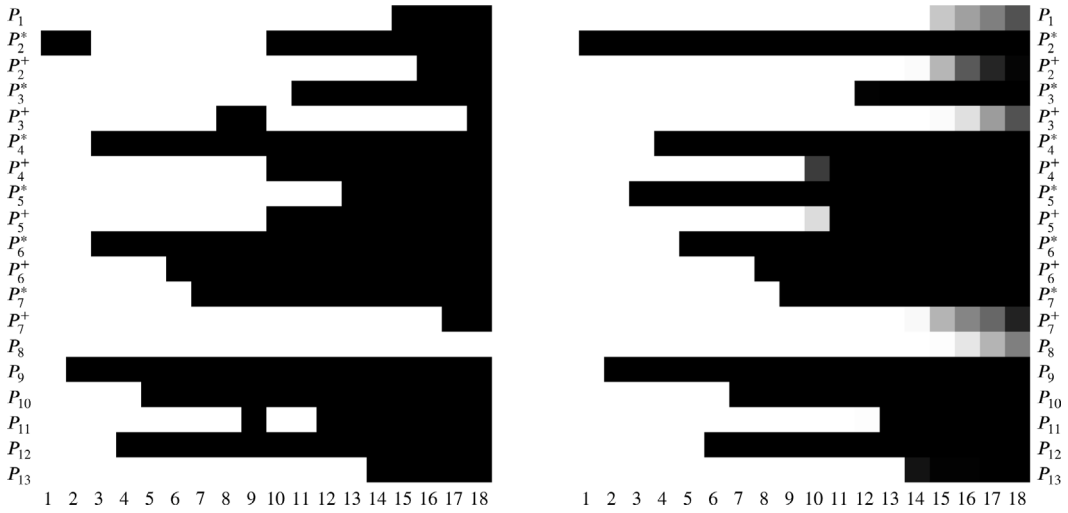

Fig. 3. Visualization of feature selection frequencies for group-CCLS (left) and SFS (right) for each feature and number of features $s = 1, \ldots, 18$. Frequency values range from 0 (0 %, white) to 1 (100 %, black) and are depicted as corresponding grayscale boxes.

5. Concluding remarks

With this work, we provide a publicly available sourcing method and data set with various attraction parameters and route data for the European passenger air transport network, and assess several models to estimate origin–destination air passenger demand on this data. Actual demand or detailed origin–destination travel data is generally proprietary and expensive to obtain. Moreover, previous works introducing air passenger volume estimation models for Europe and other markets are not directly comparable due to intransparency of the purchased data that is used for training and also needed to apply the respective models on unseen data. To overcome these drawbacks and foster research reproducibility, our open-source data set and code are intended, on the one hand, to serve as a reliable benchmark for use in future work involving European origin–destination air travel demand estimation, and on the other hand, as an easily expandable framework for data collection and curation as well as implementation and comparison of prediction models.

Moreover, our comparison of various such models demonstrates that the long-standing traditional approach to combine a basic gravity model with ordinary least-squares regression on log-linearized equations is inferior to modern machine learning techniques involving kernelization or, in particular, neural networks. Our kernelized support vector regression and neural network model also clearly outperform the Poisson pseudo-maximum-likelihood estimator that had been introduced earlier as a way to overcome some of the (statistical) drawbacks of the OLS approach. Overall, our feed-forward neural network model (four layers with tanh or ReLU activations) achieved the best cross-validated estimation accuracy by quite a large margin, improving over the other estimators by from around 15 % up to 186 % for all performance scores; we validated the superiority of the neural network model over all other approaches by statistical tests that yielded very high confidence. This conclusion—that a neural network outperforms more standard approaches—should be of interest also to practitioners with access to proprietary demand data, who may have less need for a freely available data set per se, as it shows a way to improve estimation accuracy in general. Furthermore, since methods as such obviously do not depend on the data source, our data set and collection methodology could easily be combined with or extended by route-based or origin–destination air passenger volume data from other sources than Eurostat, including, e.g., purchasable market data information tapes. It can also make sense to calibrate separate and refined models on smaller market segments; whether this is appropriate depends on the concrete practical application (such as, e.g., airline-specific revenue management), and thus should be decided on a case-by-case basis in future work.

While we took care to collect sensible choices of attraction parameters and prediction models, the results achieved here can likely be further improved. On the data side, one could, for instance, incorporate additional attraction parameters or refine the existing ones. Further variables that may affect air travel demand could be, e.g., car travel times, per-capita real disposable income, adult population numbers or measures of linguistic similarity, or indicators for multi-airport cities or whether the two countries connected by a certain route share a direct border. However, already with the choices considered in the present work, the inclusion of more than about 10 features had a marginal impact on the gravity model accuracy (regardless of the calibration method), and only the neural network model seemed to be able to really benefit. Given the amount of thought that has been put into causal models for air passenger demand over the years, it appears highly unlikely that a feature that would provide another large improvement for classical approaches has been overlooked, although it remains possible especially when focusing on smaller, more homogeneous markets as alluded to earlier. Moreover, generally, one should always utilize the best accessible data, so in particular, practitioners could extend our code to integrate proprietary origin–destination data and extend models by, e.g., service-related parameters according to their specific needs and applications. As for refinements to existing parameters, one could, e.g., take population numbers from sources that report at a finer resolution than NUTS-3 regions; from Eurostat, such a possibility may be given by the local administrative units (LAU), though we did not check data coverage across Europe for those. Presently, we use yearly passengers-on-board data, but note that Eurostat also provides this data on a monthly scale—in fact, it might happen that data for some months is missing, distorting the reported yearly numbers; we plan to expand our code to automatically handle such cases in the future. Careful manual adjustments to potentially problematic or undesired data allocations are also conceivable. In particular, the main airport of a city is sometimes located considerably far away, which can result in different NUTS-3 regions for the city and its main airport, thereby unintentionally distorting the associated attraction parameters and, ultimately, estimation model training and accuracy. Examples in our data set where we noticed such potentially "misleading" data points are Oslo (Norway) and Groningen (Netherlands). Moreover, to increase the number of data points available for model calibration, it might also be possible to carefully impute missing data, especially for data points that are almost complete in the sense that only very few attraction parameter values are missing. For example, Eurostat data does not

include NUTS-3-level GDP data for Iceland; reasonable replacement estimates for directly available data could, in this case, perhaps be obtained by combining the NUTS-3 population values from Eurostat and per-capita GDP values for Iceland from other sources. In this work, however, we did not manually manipulate the corresponding data points in order to leave the automated processing pipeline provided by our code intact.

The feature selection experiments showed that such decisions depend a lot on performance scores used to evaluate results and the motivation for feature reduction: statistical reasons, model complexity, interpretability, consistency or robustness, and the like. While the different approaches we considered all yielded different feature subset suggestions, the resulting quality measures of all correspondingly reduced models—including those not based on the log–log gravity equations (2)—only became tolerably worse. For the future, it would certainly make sense to perform feature selection directly with the model one actually wishes to deploy; in particular, incorporating a feature selection mechanism into a neural network model seems advisable. Nevertheless, as a quick and easily applicable tool, we proved the sparse regression, or cardinality-constrained least-squares model (group-CCLS) to be a viable alternative to classic OLS-based approaches such as interpreting p-values or relying on a stepwise scheme: Its variable selection was more consistent than for stepwise schemes (even without access to actual cross-validation scores), reduced-model accuracies were superior, and it can also be used for automatic feature design by easy integration of SOS1-constraints. Note that such sets of SOS1-constraints for route-based features derived from city-based parameters can straightforwardly be extended to cover further functional forms, i.e., besides product and sum features considered in the present paper, one could extend the selection to arbitrary other feature definitions and let the solver determine the best combination of all such choices.

In terms of estimation models, improvements may be possible by means of more sophisticated techniques or different components such as other kernel functions for kernelized ridge or support vector regression or different activation functions for the neural network. In particular, the architecture, i.e., the structural layout of the neural network model itself has also not yet been optimized for the task at hand. As mentioned earlier, we tried out several different neural networks to find one that works reasonably well, but with more computational effort and dedicated tools from neural architecture search (cf., e.g., Elsken et al., 2019), one can likely find network designs that further reduce overfitting and lead to even better results. Thus, a more rigorous search for a suitable neural network architecture should arguably precede the direct application of demand estimates obtained from the exemplary neural network used in the spirit of a proof-of-concept in this paper, and thus constitutes important future work. Moreover, it would be interesting to extend the neural network model to a hybrid model that combines historical time series data for different causal variables in order to directly make forecasts (rather than needing forecasted attraction parameter values to produce estimates for a future point in time), similarly to what has been done in, e.g., BaFail (2004) and Sun et al. (2019). Note that our data collection procedure could straightforwardly be extended to extract parameter data for multiple years from the external source files.

Moreover, even for the basic gravity model, some slight adjustments can yield a more fine-grained model with increased expressiveness and, likely, performance. For instance, consider using indicator variables *within exponents*: Suppose that some attraction parameter $P_{ij} = P_i P_j$ and categorical parameter I_{ij} occur in the gravity model (1) as $P_{ij}^{\alpha} I_{ij}^{\beta}$, and hence in the logarithmized formula (2) as $\alpha \log(P_{ij}) + \beta \log(I_{ij}) = \alpha \log(P_i) + \alpha \log(P_j) + \beta \log(I_{ij})$, assuming the indicator I_{ij}, like P_{ij}, is linked to the *pair* (i, j), as those used in our and most existing gravity models. Oftentimes, however, indicators can also be linked more directly to a single city, and the route-based indicators are derived from that—for instance, our parameters $P_k(i, j)$ for $k \in \{11, 12, 13\}$, see Section 2.1. Thus, we may use city-based indicators (say, I_i and I_j) more directly and replace $P_{ij}^{\alpha} I_{ij}^{\beta}$ by $P_i^{\alpha+\beta I_i} P_j^{\alpha+\beta I_j}$, which yields the logarithmized formula $(\alpha+\beta I_i)\log(P_i)+(\alpha+\beta I_j)\log(P_j)$. Then, similarly to a directed gravity model, the contribution of each attraction parameter P_i, P_j can be boosted or dampened individually depending on the respective value of the city-based indicators I_i, I_j. For instance, I_i could be 1 if i is, say, a capitol city. A route-based version of this indicator would intuitively be defined as 1 if one of the cities in the pair (i, j) is a capitol, and is hence insensitive to the situation where both of them are, which could be expected to lead to even more demand. The more general formula can actually account for such situations, while the basic gravity model formulation cannot. Analogously modified features can, of course, also be employed in the neural network model. Thus, it will be interesting to explore the capabilities of such extended model formulations in future work.

Although simplifying reality to some extent, prediction models for air passenger demand or similar quantities are valuable tools for, e.g., airline route network optimization, complementing recent related efforts for other parts of the air traffic system such as the generic aircraft operating cost functions from Förster et al. (2022). Combined with reproducibility and open-sourced data sets and collection codes, we believe they indeed offer a reasonable and principled way to move from scenario-based simulation analysis for decision support to actual optimization in several important air transport system planning problems, and improve, or even enable to begin with, comparability of results.

CRediT authorship contribution statement

Andreas M. Tillmann: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing. **Imke Joormann:** Conceptualization, Software, Validation, Data curation, Writing – review & editing. **Sabrina C.L. Ammann:** Methodology, Software, Validation, Formal analysis, Investigation, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge the funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2163/1- Sustainable and Energy Efficient Aviation – Project-ID 390881007.

Appendix A. Paired t-tests for neural net superiority

For method $m \in \{\text{direct OLS, OLS, LAV, PPML, KRR, SVR, NN}\}$ and route $k \in [N]$ (taking the role of a sample), let $d_k^m := (y^k - \hat{y}_m^k)^2$, where y^k and $\hat{y}_m^k$ are the true (known) and estimated passenger numbers, respectively. Note that with $N = 3981$, we have a fairly large number of samples to work with. Thus, we can test whether method m_1 is superior to method m_2 by formulating a t-test with null hypothesis (H_0) "the true mean of $\delta^{m_1,m_2} := d^{m_1} - d^{m_2}$ is nonnegative", hoping to reject H_0 at some significance level α in favor of the alternative $\delta_{m_1,m_2} < 0$, i.e., that the mean squared errors with respect to the ground-truth passenger volumes are better for m_1 than for m_2.

For $m_1 = \text{NN}$ and m_2 each of the other methods in turn, we thus compute the respective sample difference mean

$$\bar{D} := \frac{1}{N} \sum_{k=1}^{N} \delta_k^{\text{NN},m_2}$$

Table 8
Results of left-tailed paired t-tests w.r.t. MSE differences for neural network versus each of the other estimation methods (based on 10× repeated 10-fold cross-validation).

	(NN, Dir.OLS)	(NN, OLS)	(NN, LAV)	(NN, PPML)	(NN, KRR)	(NN, SVR)
t-stat.	−7.74344	−7.54310	−7.57160	−8.16184	−5.76350	−5.72108

and standard deviation

$$s_D := \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}\left(\delta_k^{\mathrm{NN},m_2} - \bar{D}\right)^2},$$

respectively, where $\delta_k^{\mathrm{NN},m_2} := d_k^{\mathrm{NN}} - d_k^{m_2}$. The corresponding test statistic is then given by $t = \sqrt{N}\bar{D}/s_D$, and for a given significance level α', the null hypothesis that the true mean difference of the random variables $X = (y - \hat{y}_{\mathrm{NN}})^2$ and $Y = (y - \hat{y}_{m_2})^2$ is $\mu_X - \mu_Y \geq 0$ can be rejected if t does not exceed the (left) critical value $-t_{1-\alpha';N-1}$. To account for multiple comparison effects, also known as spillage of confidence, we apply the Bonferroni correction (cf., e.g., James et al., 2021) to discount significance levels of the individual tests in order to retain a desired overall confidence for the combined hypothesis/alternative: To conclude that "the NN model is superior to all others" (as the alternative to "some model is at least as good as the neural net") at level α, we can employ a significance level $\alpha' \leq \alpha/6$ in each of the six individual tests. Thus, to assert an overall significance level below 0.000 1 %, it suffices to employ, say, $\alpha' = 0.00000016$. The following Table 8 lists the test statistics for each pair (NN, m_2), where all calibrated models employed for the estimation were chosen as those with lowest respective mean validation loss among 10× repeated 10-fold cross-validation runs[3]; the critical value is $-t_{0.99999984;3980} \approx -5.11999$. Clearly, in all individual difference tests, we can reject the respective null hypothesis and, via the Bonferroni method, can therefore conclude that our neural network model indeed significantly outperforms all other methods considered here for the estimation of air passenger demand on our data set, at an overall significance level below 0.000 1 % and with the (type-I error) probability that no null hypothesis was rejected despite actually being true exceeding 99.999 9 %. Indeed, all t-statistics are well below the critical value, so the tests provide strong evidence of significant superiority of the neural network approach.

Appendix B. Feature selection details

In this appendix, we provide the details on the feature selection results summarized in Section 4.2.2. Recall that we consider OLS-based feature selection by means of the two stepwise regression schemes sequential forward selection (SFS) and recursive feature elimination (RFE), using cross-validation scores, as well as the modified cardinality-constrained least-squares model (group-CCLS). To begin, we also consider the common basic selection based on p-values and associated statistical significance.

For the variables in the log–log gravity model (2) with all features, trained by standard OLS using all available data, it turns out that seven of nineteen parameters appear insignificant at the 5 % level—P_1 (with p-value 0.225), P_2^+ (0.151), P_3^* (0.251), P_3^+ (0.351), P_5^* (0.058), P_7^+ (0.282) and P_8 (0.567)—and two more, P_7^* (0.048) and P_{13} (0.036), are weakly significant, i.e., significant at the 5 % level but not at the 1 % level. All remaining variables show stronger evidence of significance, with p-values clearly below 0.01. At the 0.1 % level, one more variable would be deemed insignificant, namely P_5^+ with p-value 0.003. The least conservative threshold choice thus suggest keeping 12 features in the model: P_2^*, P_4^*, P_4^+, P_5^+, P_6^*, P_6^+, P_7^*, P_9, P_{10}, P_{11}, P_{12} and P_{13}. Let us keep this in mind for now, and next consider the algorithmic feature selection methods.

[3] Note that the comparison here thus includes estimations for all data points both used and unseen during model training, though the respective data splits were generally different.

The SFS implementation in scikit-learn asks to specify the number of features s to be selected in advance. Similarly, an optimal (group-)CCLS solution will always actively use s features, provided the data columns are linearly independent (as is the case in our data set), since every additional feature allows to further reduce the least-squares objective. The RFE functionality in scikit-learn, however, only allows to specify a minimum number of features to be chosen, and automatically determines the actual number. Therefore, we will analyze the results of 100 repetitions using random 10-fold CV-like data splits for the three methods as follows: First, we compare the behavior of group-CCLS and SFS for all values of s up to $K-1 = 18$. Then, we will assess the outcome of RFE and compare to the results from group-CCLS and SFS both with respect to the same number of features as suggested by RFE as well as the respective number of features beyond which no significant further accuracy gains were attainable according to group-CCLS or SFS. The final cross-validation performance results of the models restricted to the deduced "best" feature subsets, calibrated with all methods, was already discussed in the main part of the paper, see Section 4.2.2 (in particular, Table 5).

Table 9 summarizes the performance scores for group-CCLS and SFS; note that, while both methods (as well as RFE later) make use of the CV data split to determine their respective feature selection, the scores reported are obtained from OLS-calibration of the correspondingly reduced log–log gravity models using all available data. To save space, we omit the MAD values, which behaved analogously to the earlier experiments (cf. Section 4.2.1) and showed no unexpected variations.

The results in Table 9 clearly corroborate our earlier claim that stepwise regression offers no advantage over the CCLS approach, at least in our application: Even without access to the actual cross-validation scores, the exact sparse regression approach outperforms SFS for all s with respect to most or even all quality scores. If only one or two features are allowed, both methods identify the same ones, P_2^* and $\{P_2^*, P_9\}$, respectively. For $s \geq 13$, the scores for both methods hardly improve with additional features and are nearly identical, differing only in the fourth decimal. Thus, SFS appears inferior to group-CCLS, particularly in the arguably more interesting regime where notably fewer features are included in the model, while the concrete selection does not make much difference any more if most features are retained.

Moreover, since we used random data splits as in CV, the features selected by either method may vary across folds for each s. Then, for the actual feature selection for a specific s, it is sensible to choose the variables that were selected most often. Therefore, it is desirable that a feature selection method is *consistent* in the sense that the selection frequencies ideally give clear indications which variables to pick. As visualized in Fig. 3, SFS yields notably less consistent parameter selection frequencies than group-CCLS—observe that in several columns (with $s \geq 10$), SFS leads to gray fields, mirroring larger uncertainty as to which variable to choose, whereas group-CCLS exhibits no such uncertainty at all. This further supports the sentiment voiced in, e.g., Smith (2018) that stepwise regression is unreliable and should be avoided. From Fig. 3, we can also see that once a variable enters the model, SFS never discards it again and indeed, adds it with increasing frequency for growing s, whereas group-CCLS reconfigures parameter choices anew for each s and can, consequently, exclude a variable again that was used for a smaller s. This reflects the greedy nature of SFS versus the combinatorial freedom in the CCLS approach.

As a criterion to determine which features a method actually suggests to be retained in a reduced model, we opt to identify the "sweet spot", i.e., the number of features beyond which no relevant further improvements in model accuracy can be achieved. Given that we use

Table 9
Result summary for feature selection methods group-CCLS and SFS based on 100× repeated 10-fold CV, for $s = 1, \ldots, 18$ of $K = 19$ features.

	group-CCLS				SFS			
s	R^2	R^2 (log.)	MAE	MAPE	R^2	R^2 (log.)	MAE	MAPE
1	0.1223	0.2354	125872	0.9357	0.1223	0.2354	125872	0.9357
2	0.1700	0.2750	123102	0.9181	0.1700	0.2750	123102	0.9181
3	0.2253	0.3263	118654	0.8864	0.1171	0.3007	123933	0.8971
4	0.2365	0.3421	118286	0.8907	0.1787	0.3227	120618	0.8846
5	0.2001	0.3511	119241	0.8758	0.1987	0.3349	119391	0.8762
6	0.2042	0.3596	118692	0.8644	0.2137	0.3471	118957	0.8815
7	0.2130	0.3657	118192	0.8567	0.2108	0.3565	118394	0.8673
8	0.2181	0.3694	117699	0.8530	0.2167	0.3659	117790	0.8545
9	0.2203	0.3735	117373	0.8469	0.2206	0.3713	117529	0.8504
10	0.2281	0.3765	116907	0.8441	0.2251	0.3749	117134	0.8462
11	0.2328	0.3786	116611	0.8417	0.2317	0.3784	116707	0.8437
12	0.2330	0.3807	116546	0.8385	0.2363	0.3804	116417	0.8413
13	0.2366	0.3828	116351	0.8380	0.2366	0.3828	116351	0.8380
14	0.2376	0.3833	116277	0.8370	0.2376	0.3833	116282	0.8371
15	0.2386	0.3836	116275	0.8370	0.2380	0.3835	116297	0.8372
16	0.2385	0.3838	116330	0.8376	0.2382	0.3837	116315	0.8374
17	0.2387	0.3840	116332	0.8376	0.2383	0.3839	116316	0.8375
18	0.2385	0.3841	116308	0.8376	0.2384	0.3840	116315	0.8376

Table 10
Result summary for feature selection method RFE based on 100× repeated 10-fold CV.

s	freq	R^2	R^2 (log.)	MAE	MAPE
16	28 %	0.2369	0.3837	116331	0.8380
17	2 %	0.2377	0.3839	116312	0.8376
18	3 %	0.2385	0.3841	116308	0.8376
19	67 %	0.2386	0.3841	116311	0.8376

several performance scores, we propose to decide based on the first time that all five cross-validation scores—including MAD scores, which were not reported in Table 9—improve and the average improvement over all scores drops below 1 %. For group-CCLS, this happens when moving from $s = 10$ to $s = 11$, which we may thus take to mean that 10 features yield a sufficiently accurate model. From Fig. 3, we can glean that the attraction parameters selected by this CCLS interpretation are P_2^*, P_4^*, P_4^+, P_5^+, P_6^*, P_6^+, P_7^*, P_9, P_{10} and P_{12}. This selection is as consistent as possible, with each feature being selected every time in the repeated runs with different random data splits each. We also note (cf. Fig. 3) that, interestingly, the same-currency indicator P_8 is *never* chosen by group-CCLS for any $s < 19$.

Applying the same criterion to the SFS results, the 11 features P_2^*, P_4^*, P_4^+, P_5^*, P_5^+, P_6^*, P_6^+, P_7^*, P_9, P_{10}, and P_{12} are chosen, also all with selection frequencies 100 %. Note that this corresponds to the selection by group-CCLS but additionally includes P_5^*. For $s = 10$, SFS exhibited some uncertainty, choosing P_5^+ 17 % and P_4^+ 83 % of times, whereas group-CCLS always optimally chose the latter.

Finally, we will now discuss the third feature selection method, RFE. We ran the RFE experiments without requiring a restrictive minimum number of features; indeed, recall that RFE determines the final number of features automatically. As it turned out, RFE never retained fewer than 16 features, and did not eliminate any at all in the majority of cases. The results are summarized in Table 10 analogously to Table 9, except that column s now corresponds to the feature number produced by RFE, with the percentages of experiments (among the 100 repetitions with 10-fold CV data splits) with the respective outcomes reported in the column labeled "freq".

Since RFE proceeds by iteratively removing features instead of adding them, and the average *deterioration* percentage across all scores is below 1 % for each removal of a feature, our earlier selection criterion is inapplicable here. Taking overall consistency into account, 16 features were always included; the remaining ones have selection frequencies at most 72 %. Thus, we may interpret the RFE results to suggest retaining 16 parameters: all except P_1, P_3^* and P_8.

For the summary of selected feature subsets and the evaluation of the correspondingly reduced model accuracies achievable with the various calibration methods, see Tables 4 and 5, respectively, and the discussion in Section 4.2.2.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. URL https://www.tensorflow.org.

Adler, N., Njoya, E.T., Volta, N., 2018. The multi-airline p-hub median problem applied to the African aviation market. Transp. Res. A 107, 187–202. http://dx.doi.org/10.1016/j.tra.2017.11.011.

Airports Commission, 2013. Aviation Demand Forecasting. Research Report, URL https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/73143/aviation-demand-forecasting.pdf.

Alekseev, K.P.G., Seixas, J.M., 2009. A multivariate neural forecasting modeling for air transport — Preprocessed by decomposition: A Brazilian application. J. Air Transp. Manag. 15, 212–216. http://dx.doi.org/10.1016/j.jairtraman.2008.08.008.

Anderson, J.E., 2011. The gravity model. Annu. Rev. Econ. 3, 133–160. http://dx.doi.org/10.1146/annurev-economics-111809-125114.

Anderson, N.H., Shanteau, J., 1977. Weak inference with linear models. Psychol. Bull. 84, 1155–1170. http://dx.doi.org/10.1037/0033-2909.84.6.1155.

Angelova, D., Blanco Lupio, N., 2020. Constructing a meteorological indicator dataset for selected European NUTS 3 regions. Data Brief 31, 105786. http://dx.doi.org/10.1016/j.dib.2020.105786.

Aydın, U., Ülengin, B., 2022. Analyzing air cargo flows of Turkish domestic routes: A comparative analysis of gravity models. J. Air Transp. Manag. 102, 102217. http://dx.doi.org/10.1016/j.jairtraman.2022.102217.

BaFail, A.O., 2004. Applying data mining techniques to forecast number of airline passengers in Saudi Arabia (domestic and international travels). J. Air Transp. 9, 100–115.

Baier, F., Berster, P., Gelhausen, M., 2022. Global cargo gravitation model: Airports matter for forecasts. Int. Econ. Econ. Policy 19, 219–238. http://dx.doi.org/10.1007/s10368-021-00525-2.

Becker, K., Terekhov, I., Niklaß, M., Gollnick, V., 2018. A global gravity model for air passenger demand between city pairs and future interurban air mobility markets identification. In: Proc. AAIA/Aviation Forum. American Institute of Aeronautics and Astronautics, Inc., http://dx.doi.org/10.2514/6.2018-2885.

Berk, R.A., 2004. Regression Analysis: A Constructive Critique. In: Advanced Quantitative Techniques in the Social Sciences, vol. 11, SAGE Publications Inc..

Bestuzheva, K., Besançon, M., Chen, W.-K., Chmiela, A., Donkiewicz, T., van Doornmalen, J., Eifler, L., Gaul, O., Gamrath, G., Gleixner, A., Gottwald, L., Graczyk, C., Halbig, K., Hoen, A., Hojny, C., van der Hulst, R., Koch, T., Lübbecke, M., Maher, S.J., Matter, F., Mühmer, E., Müller, B., Pfetsch, M.E., Rehfeldt, D., Schlein, S., Schlösser, F., Serrano, F., Shinano, Y., Sofranac, B., Turner, M., Vigerske, S., Wegscheider, F., Wellner, P., Weninger, D., Witzig, J., 2021. The SCIP Optimization Suite 8.0. Tech. Rep., Optimization Online, URL http://www.optimization-online.org/DB_HTML/2021/12/8728.html.

Bilotkach, V., Pejcinovska, M., 2012. Distribution of airline tickets: A tale of two market structures. In: Peoples, J. (Ed.), Pricing Behavior and Non-Price Characterictics in the Airline Industry. In: Advances in Airline Economics, vol. 3, Emerald Group Publishing Ltd., pp. 107–138. http://dx.doi.org/10.1108/S2212-1609(2011)0000003007.

Birnbaum, M.H., 1973. The devil rides again: Correlation as an index of fit. Psychol. Bull. 79, 239–242. http://dx.doi.org/10.1037/h0033853.

Brown, S.L., Watkins, W.S., 1968. The demand for air travel: A regression study of time-series and cross-sectional data in the U.S. domestic market. Highw. Res. Rec. 21–34.

Carmona-Benítez, R.B., Nieto, M.R., 2020. SARIMA damp trend grey forecasting model for airline industry. J. Air Transp. Manag. 82, 101736. http://dx.doi.org/10.1016/j.jairtraman.2019.101736.

Chang, C.-C., Lin, C.-J., 2011. LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2, 27. http://dx.doi.org/10.1145/1961189.1961199.

Chollet, F., et al., 2015. Keras. https://keras.io.

Cleophas, C., Frank, M., Kliewer, N., 2009. Recent developments in demand forecasting for airline revenue management. Int. J. Revenue Manag. 3, 252–269.

Colquhoun, D., 2014. An investigation of the false discovery rate and the misinterpretation of *p*-values. R. Soc. Open Sci. 1, 140216. http://dx.doi.org/10.1098/rsos.140216.

Cook, A.J., Kluge, U., Paul, A., Cristobal, S., 2017. Factors influencing European passenger demand for air transport. In: Proc. Air Transport Research Society World Conference.

Dielman, T.E., 2005. Least absolute value regression: Recent contributions. J. Stat. Comput. Simul. 75, 263–286. http://dx.doi.org/10.1080/0094965042000223680.

Dobruszkes, F., 2013. The geography of European low-cost airline networks: A contemporary analysis. J. Transp. Geogr. 28, 75–88. http://dx.doi.org/10.1016/j.jtrangeo.2012.10.012.

Doganis, R., 2019. Flying Off Course: Airline Economics and Marketing, fifth ed. Routledge.

Elsken, T., Metzen, J.H., Hutter, F., 2019. Neural architecture search: A survey. J. Mach. Learn. Res. 20, 1997–2017.

Eurostat, 2022. Your key to European statistics. URL https://ec.europa.eu/eurostat/data/database.

Eurostat (European Commission), 2021. Eurostat Regional Yearbook: 2021 Edition. Publications Office of the European Union, Luxembourg, http://dx.doi.org/10.2785/894358,

Ferri, F.J., Pudil, P., Hatef, M., Kittler, J., 1994. Comparative study of techniques for large-scale feature selection. Mach. Intell. Pattern Recognit. 16, 403–413. http://dx.doi.org/10.1016/B978-0-444-81892-8.50040-7.

Fildes, R., Wei, Y., Ismail, S., 2011. Evaluating the forecasting performance of econometric models of air passenger traffic flows using multiple error measures. Int. J. Forecast. 27, 902–922. http://dx.doi.org/10.1016/j.ijforecast.2009.06.002.

Förster, P., Yildiz, B., Feuerle, T., Hecker, P., 2022. Approach for cost functions for the use in trade-off investigations assessing the environmental impact of a future energy efficient European aviation. Aerospace 9, 167. http://dx.doi.org/10.3390/aerospace9030167.

Gelhausen, M.C., Berster, P., 2017. A gravity model for estimating passenger origin-destination flows between countries worldwide. URL https://elib.dlr.de/113857/.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, M. (Eds.), Proc. 13th International Conference on Artificial Intelligence and Statistics. In: Proceedings of Machine Learning Research, 9, pp. 249–256.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, http://www.deeplearningbook.org.

Grimme, W., Maertens, S., 2019. Flightpath 2050 revisited — An analysis of the 4-hour-goal using flight schedules and origin-destination passenger demand data. Transp. Res. Procedia 43, 147–155. http://dx.doi.org/10.1016/j.trpro.2019.12.029.

Grosche, T., 2009. Computational Intelligence in Integrated Airline Scheduling. In: Studies in Computational Intelligence, vol. 173, Springer, http://dx.doi.org/10.1007/978-3-540-89887-0.

Grosche, T., Rothlauf, F., Heinzl, A., 2007. Gravity models for airline passenger volume estimation. J. Air Transp. Manag. 13, 175–183. http://dx.doi.org/10.1016/j.jairtraman.2007.02.001.

Gurobi Optimization, LLC, 2022. Gurobi optimizer reference manual. URL https://www.gurobi.com.

Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. Mach. Learn. 46, 389–422. http://dx.doi.org/10.1023/A:1012487302797.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning. Data Mining, Inference, and Prediction, second ed. In: Springer Series in Statistics, Springer.

Hazledine, T., 2017. An augmented gravity model for forecasting passenger air traffic on city-pair routes. J. Transp. Econ. Policy 51, 208–224.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing humal-level performance on ImageNet classification. In: Proc. IEEE International Conference on Computer Vision. IEEE, pp. 1024–1034. http://dx.doi.org/10.1109/ICCV.2015.123.

Hoffman, T., Schölkopf, B., Smola, A.J., 2008. Kernel methods in machine learning. Ann. Stat. 36, 1171–1220. http://dx.doi.org/10.1214/009053607000000677.

Huang, Z., Wu, X., Garcia, A.J., Fik, T.J., Tatem, A.J., 2013. An open-access modeled passenger flow matrix for the global air network in 2010. PLoS One 8, e64317. http://dx.doi.org/10.1371/journal.pone.0064317.

IBM ILOG, 2022. CPLEX optimization studio. URL https://www.ibm.com/products/ilog-cplex-optimization-studio.

International Civic Aviation Organization, 2006. Manual on air traffic forecasting. URL https://www.icao.int/MID/Documents/2014/Aviation%20Data%20Analyses%20Seminar/8991_Forecasting_en.pdf.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2021. An Introduction to Statistical Learning, second ed. In: Springer Texts in Statistics, Springer.

Jin, F., Li, Y., Sun, S., Li, H., 2020. Forecasting air passenger demand with a new hybrid ensemble approach. J. Air Transp. Manag. 83, 101744. http://dx.doi.org/10.1016/j.jairtraman.2019.101744.

Joormann, I., Tillmann, A.M., Ammann, S.C.L., 2023. airPaDE. URL https://github.com/imkejoor/airPaDE.

Jorge-Calderón, J.D., 1997. A demand model for scheduled airline services on international European routes. J. Air Transp. Manag. 3, 23–35. http://dx.doi.org/10.1016/S0969-6997(97)82789-5.

Kabir, M., Salim, R., Al-Mawali, N., 2017. The gravity model and trade flows: Recent developments in econometric modeling and empirical evidence. Econ. Anal. Policy 56, 60–71. http://dx.doi.org/10.1016/j.eap.2017.08.005.

Kanafani, A.K., 1983. Transportation Demand Analysis. McGraw-Hill, New York.

Kanji, G.K., 2006. 100 Statistical Tests, third ed. SAGE Publications Inc..

Kepaptsoglou, K., Karlaftis, M.G., Tsamboulas, D., 2010. The gravity model specification for modeling international trade flows and free trade agreement effects: A 10-year review of empirical studies. Open Econ. J. 3, 1–13. http://dx.doi.org/10.2174/1874919401003010001.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv:1412.6980 [cs.LG] / ICLR 2015.

Lehmann, E.L., Romano, J.P., 2005. Testing Statistical Hypotheses, third ed. In: Springer Texts in Statistics, Springer.

Mao, L., Wu, X., Huang, Z., Tatem, A.J., 2015. Modeling monthly flows of global air travel passengers: An open-access data resource. J. Transp. Geogr. 48, 52–60. http://dx.doi.org/10.1016/j.jtrangeo.2015.08.017.

McFadden, D., 1974. Conditional logit analysis of qualitative choice behavior. In: Zarembka, P. (Ed.), Frontiers in Econometrics. Academic Press, pp. 105–142.

Motta, V., 2019. Estimating Poisson pseudo-maximum-likelihood rather than log-linear model of a log-transformed dependent variable. RAUSP Manag. J. 54, 508–518. http://dx.doi.org/10.1108/RAUSP-05-2019-0110.

Murphy, K.P., 2012. Machine Learning: A Probabilistic Perspective. Adaptive Computation and Machine Learning, MIT Press.

Nõmmik, A., Kukemeld, S., 2016. Developing gravity model for airline regional route modelling. Aviation 20, 32–37. http://dx.doi.org/10.3846/16487788.2016.1168007.

Oesingmann, K., 2022. The effect of the European emissions trading system (EU ETS) on aviation demand: An empirical comparison with the impact of ticket taxes. Energy Policy 160, 112657. http://dx.doi.org/10.1016/j.enpol.2021.112657.

O'Kelly, M.E., Song, W., Shen, G., 1995. New estimates of gravitational attraction by linear programming. Geogr. Anal. 27, 271–285. http://dx.doi.org/10.1111/j.1538-4632.1995.tb00911.x.

Park, Y., O'Kelly, M.E., 2016. Origin–destination synthesis for aviation network data: Examining hub operations in the domestic and international US markets. J. Adv. Transp. 50, 2288–2305. http://dx.doi.org/10.1002/ATR.1459.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830, URL https://scikit-learn.org.

Rengaraju, V.R., Thamizh Arasan, V., 1992. Modeling for air travel demand. J. Transp. Eng. 118, 371–380. http://dx.doi.org/10.1061/(ASCE)0733-947X(1992)118:3(371).

Rousseeuw, P.J., Leroy, A.M., 1987. Robust Regression and Outlier Detection. In: Wiley Series in Probability and Statistics, John Wiley & Sons.

Russon, M.G., 1990. Iterative nonlinear estimation of air passenger flow sensitivity to political boundaries and a complex function of distance. Logist. Transp. Rev. 26, 323+.

Russon, M.G., Riley, N.F., 1993. Airport substitution in a short haul model of air transportation. Int. J. Transp. Econ. 20, 157–174.

Santos Silva, J.M.C., Tenreyro, S., 2006. The log of gravity. Rev. Econ. Stat. 88, 641–658. http://dx.doi.org/10.1162/rest.88.4.641.

Santos Silva, J.M.C., Tenreyro, S., 2022. The log of gravity at 15. Port. Econ. J. http://dx.doi.org/10.1007/s10258-021-00203-w.

Schölkopf, B., Smola, A.J., 2001. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and beyond. MIT Press.

Schultz, R.L., 1972. Studies of airline passenger demand: A review. Transp. J. 11, 48–62.

Seabold, S., Perktold, J., 2010. Statsmodels: Econometric and statistical modeling with python. In: Proceedings of the 9th Python in Science Conference, vol. 57, pp. 92–96, URL https://www.statsmodels.org/.

Shmueli, G., 2010. To explain or to predict? Statist. Sci. 25, 289–310. http://dx.doi.org/10.1214/10-STS330.

Sivrikaya, O., Tunç, E., 2013. Demand forecasting for domestic air transportation in Turkey. Open Transp. J. 7, 20–26. http://dx.doi.org/10.2174/1874447820130508001.

Smith, G., 2018. Step away from stepwise. J. Big Data 5, 32. http://dx.doi.org/10.1186/s40537-018-0143-6.

Smola, A.J., Schölkopf, B., 2004. A tutorial on support vector regression. Stat. Comput. 14, 199–222. http://dx.doi.org/10.1023/B:STCO.0000035301.49549.88.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1929–1958.

Suau-Sanchez, P., Voltes-Dorta, A., Rodríguez-Déniz, H., 2017. An assessment of the potential for self-connectivity at European airports in holiday markets. Tour. Manag. 62, 54–64. http://dx.doi.org/10.1016/j.tourman.2017.03.022.

Sun, S., Lu, H., Tsui, K.-L., Wang, S., 2019. Nonlinear vector auto-regression neural network for forecasting air passenger flow. J. Air Transp. Manag. 78, 54–62. http://dx.doi.org/10.1016/j.jairtraman.2019.04.005.

Suryani, E., Chou, S.-Y., Chen, C.-H., 2010. Air passenger demand forecasting and passenger terminal capacity expansion: A system dynamics framework. Expert Syst. Appl. 37, 2324–2339. http://dx.doi.org/10.1016/j.eswa.2009.07.041.

Swan, W.M., Adler, N., 2006. Aircraft trip cost parameters: A function of stage length and seat capacity. Transp. Res. E 42, 105–115. http://dx.doi.org/10.1016/j.tre.2005.09.004.

Terekhov, I., 2017. Forecasting Air Passenger Demand Between Settlements Worldwide Based on Socio–Economic Scenarios (Ph.D. thesis). Technische Universität Hamburg-Harburg.

Terekhov, I., Ghosh, R., Gollnick, V., 2015. A concept of forecasting origin-destination air passenger demand between global city pairs using future socio-economic scenarios. http://dx.doi.org/10.2514/6.2015-1640, Proc. 53rd AIAA Aerospace Sciences Meeting.

Thou, H., Xia, J., Lui, Q., Nikolova, G., Sun, J., Hughes, B., Kelobonye, K., Wang, H., Falkmer, T., 2018. Investigating the impact of catchment areas of airports on estimating air travel demand: A case study of regional Western Australia. J. Air Transp. Manag. 70, 91–103. http://dx.doi.org/10.1016/j.jairtraman.2018.05.001.

Tian, H., Presa-Reyes, M., Tao, Y., Wang, T., Pouyanfar, S., Alonso, Jr., M., Luis, S., Shyu, M.-L., Chen, S.-C., Sitharama Iyengar, S., 2021. Data analytics for air travel data: A survey and new perspectives. ACM Comput. Surv. 54, 167. http://dx.doi.org/10.1145/3469028.

Tillmann, A.M., Bienstock, D., Lodi, A., Schwartz, A., 2021. Cardinality minimization, constraints, and regularization: A survey. arXiv:2106.09606 [math.OC].

Verleger, Jr., P.K., 1972. Models of the demand for air transportation. Bell J. Econ. Manag. Sci. 3, 437–457. http://dx.doi.org/10.2307/3003032.

Vinod, B., 2021. The Evolution of Yield Management in the Airline Industry. Origins to the Last Frontier. In: Management for Professionals, Springer.

Vovk, V., 2013. Kernel ridge regression. In: Schölkopf, B., Luo, Z., Vovk, V. (Eds.), Empirical Inference. Springer, pp. 105–116. http://dx.doi.org/10.1007/978-3-642-41136-6_11.

Wang, S., Gao, Y., 2021. A literature review and citation analysis of air travel demand studies published between 2010 and 2020. J. Air Transp. Manag. 97, 102135. http://dx.doi.org/10.1016/j.jairtraman.2021.102135.

Wong, C.W.H., Cheung, T.K.-Y., Zhang, A., 2023. Airport and air route demand planning: A connectivity-based methodology for new route identification. http://dx.doi.org/10.2139/ssrn.4379104, SSRN.

Xiao, Y., Liu, J.J., Hu, Y., Wang, Y., Lai, K.K., Wang, S., 2014. A neuro-fuzzy combination model based on singular spectrum analysis for air transport demand forecasting. J. Air Transp. Manag. 39, 1–11. http://dx.doi.org/10.1016/j.jairtraman.2014.03.004.

Yu, J., 2021. A new way of airline traffic prediction based on GCN-LSTM. Front. Neurorobotics 15, 661037. http://dx.doi.org/10.3389/fnbot.2021.661037.

Zhang, Y., Lin, F., Zhang, A., 2018. Gravity models in air transport research: A survey and an application. In: Blonigen, B.A., Wilson, W.W. (Eds.), Handbook of International Trade and Transportation. Edward Elgar Publishing, pp. 141–158.